

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA.

TRABAJO FIN DE GRADO

***< PECERA AUTOMATIZADA, MONITORIZADA
POR PANTALLA LCD Y POR MOVIL VIA
BLUETOOTH, BASADA EN LA
PLATAFORMA ARDUINO.>***

Alumno/Alumna: <Salas, Jiménez, Héctor Javier>

Director/Directora (1): <Oleagordia, Aguirre, Iñigo Javier>

Curso: <2019-2020>

Fecha: <28, Octubre, 2019>

- **RESUMEN**

La finalidad de este proyecto consiste en el diseño, desarrollo y posterior montaje de una pecera automatizada, capaz de realizar diversos controles sobre los factores que la rodean y que puedan comprometerla tanto internamente como externamente. Mediante una pantalla LCD, se mostrarán los distintos parámetros cada cierto tiempo. Como alternativa se podrá acceder a estos datos a través de una aplicación en el móvil vía bluetooth.

Para ello, se repartirán distintos sensores por toda la pecera, los cuales serán los encargados de recoger la información relacionada con la temperatura interna y externa, con el nivel del agua y la turbidez. Estos llevarán la información a un micro-controlador Arduino que analizará y procesará toda esta información para después enviarla a la pantalla LCD y para controlar las válvulas de entrada o salida de agua.

- **PALABRAS CLAVE**

Sensor, Pecera, Datos, Turbidez, Temperatura, Válvula, Nivel, Agua, Móvil, Pantalla, Arduino.

- **ABSTRACT**

The purpose of this project is to design, develop and subsequently assembly an automated fish tank, capable of performing various controls on the factors that surround it and that can compromise it, both internally and externally. Through an LCD screen, the different parameters will be displayed. As an alternative you can access these data through an application on your mobile phone via Bluetooth.

For this, different sensors will be distributed throughout the fish tank, which will be responsible for collecting information related to internal and external temperature, water level and turbidity. The sensors will take all the information to an Arduino microcontroller that will analyze and process all this information and then send it to the LCD screen and to control the water inlet or outlet valves.

- **KEY WORDS**

Sensor, Fishbowl, Data, Turbidity, Temperature, Valve, Level, Water, Mobile phone, Screen, Arduino.

- **LABURPENA**

Proiektu honek automatizatutako arrainontzi bat diseinatzea, garatzea eta muntatzea du helburu. Arrainontzi automatizatu honek haren inguruko zenbait faktore, funtzionamendua arriskuan jar dezaketen barneko zein kanpoko faktoreak, kontrolatu ahal izango ditu. Parametro esanguratsuak LCD pantaila batetan bistaratuko dira aldiro. Gainera, parametro horiek mugikor batetan ere bistaratu ahal izango dira, Bluetooth bidezko aplikazio baten bitartez.

Hartarako guztirako, arrainontzian zehar zenbait sentsore kokatu dira, kanpo-tenperatura, barne-tenperatura, ur-maila eta uhertasunaren inguruko informazioa bilduko dutenak. Sentsore horiek Arduino mikrokontroladore batera eroango dute informazioa. Mikrokontroladoreak informazio hori aztertu eta prozesatuko du, gerora, berau LCD pantailan bistaratzeko eta ur-irteera/sarrera balbulak kontrolatzeko.

- **GAKO HITZAK**

Sentsore, Arrnontzi, Datu, Turbidez, Tenperatura, Balbula, Maila, Ura, Telefono, Pantaila, Arduino.

ÍNDICE

1 - MEMORIA.....	7
1.1 LISTA DE FIGURAS, LISTA DE TABLAS Y LISTA DE DIAGRAMAS.....	7
1.2 CONTEXTO Y DIAGRAMAS DE BLOQUE.....	12
1.2.3 Contexto.....	12
1.2.4 Diagrama de bloques.....	13
1.3 OBJETIVOS Y ALCANCE DEL TRABAJO.....	13
1.4 DESCRIPCIÓN DE TAREAS Y PROCEDIMIENTOS.....	14
1.5 IMPLEMENTACIÓN HARDWARE.....	14
1.5.1 COMPONENTES.....	15
1.5.1.1 NTC.....	15
1.5.1.2 LM335.....	16
1.5.1.3 TERMOPAR-TIPO K.....	16
1.5.1.4 RTD-PT100.....	17
1.5.1.5 SENSOR DE ULTRASONIDOS PIEZOELECTRICO HC-SR04.....	17
1.5.1.6 ELECTRO-VÁLVULAS.....	18
1.5.1.7 SENSOR DE NIVEL DE AGUA.....	18
1.5.1.8 SENSOR DE TURBIDEZ.....	19
1.5.1.9 MÓDULO BLUETOOTH HC-06.....	20
1.5.1.10 PANTALLA LCD 20X4 CON MÓDULO I2C.....	20
1.5.2 SISTEMA DE PROCESAMIENTO.....	21
1.5.2.1 MICROCONTROLADOR ARDUINO.....	21
1.5.2.2 TIPOS DE ARDUINO.....	22
1.5.3 CÁLCULOS HARDWARE.....	23
1.5.3.1 NTC.....	23
1.5.3.2 LM335.....	29
1.5.3.3 TERMOPAR-TIPO K.....	32
1.5.3.4 RTD-PT100.....	33
1.5.3.5 SENSOR DE ULTRASONIDOS PIEZOELECTRICO HC-SR04.....	38
1.5.3.6 ELECTRO-VÁLVULAS.....	40
1.5.3.7 SENSOR DE NIVEL DE AGUA.....	41
1.5.3.8 SENSOR DE TURBIDEZ.....	41
1.5.3.9 MÓDULO BLUETOOTH HC-06.....	43
1.5.3.10 PANTALLA LCD 20X4 CON MÓDULO I2C.....	43
1.6 IMPLEMENTACIÓN SOFTWARE.....	44
1.6.1 MODULO BLUETOOTH HC-06.....	44
1.6.2 APLICACIÓN MOVIL.....	46
1.6.3 PROGRAMACIÓN CON ARDUINO.....	49
1.6.4 CÁLCULOS SOFTWARE.....	50

2. CONCLUSIONES	70
2.1 DIAGRAMA DE GANTT	71
2.2 PROBLEMAS HALLADOS	72
2.3 RESULTADOS	74
3. ASPECTOS ECONÓMICOS	80
3.1 LISTA DE MATERIALES NECESARIOS.....	80
3.1.2 LUGARES DE COMPRA	82
3.1.3 COSTES NO INCLUIDOS	82
3.1.4 DISPOSITIVOS DE COMPROBACIÓN	83
3.1.5 PRESUPUESTO FINAL.....	83
4. BIBLIOGRAFÍA	85
4.1 ANEXOS.....	88
4.1.1 PLACAS PCB	88

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA.

TRABAJO FIN DE GRADO

***< PECERA AUTOMATIZADA, MONITORIZADA
POR PANTALLA LCD Y POR MOVIL VIA
BLUETOOTH, BASADA EN LA PLATAFORMA
ARDUINO >***

DOCUMENTO 1- <MEMORIA>

Alumno/Alumna: <Salas, Jiménez, Héctor Javier>

Director/Directora (1): <Oleagordia, Aguirre, Iñigo Javier>

Curso: <2019-2020>

Fecha: < 28, Octubre, 2019>

1 - MEMORIA

1.1 LISTA DE FIGURAS, LISTA DE TABLAS Y LISTA DE DIAGRAMAS

- **LISTA DE FIGURAS**

Figura1. Diagrama de un sistema de medida.

Figura2. Diagrama de bloques de la pecera automatizada.

Figura3. Sensor NTC utilizado en el proyecto.

Figura4. Encapsulado TO-92 LM335.

Figura5. Termopar tipo K.

Figura6. Sonda Pt10.

Figura7. Funcionamiento modo eco del sensor de ultrasonidos.

Figura8. Interior y exterior de una electro-válvula.

Figura9. Sensor de nivel.

Figura10. Esquema interno del sensor de turbidez.

Figura11. Sensor de turbidez con y módulo V1.0 de conexión.

Figura12. Módulos bluetooth HC-06 (esclavo) y HC-05 (maestro/esclavo).

Figura13. Pantalla lcd 20x4 con módulo I2C.

Figura14. Distintos tipos de Arduino.

Figura15. Placa Arduino UNO y DUE.

Figura16. Esquema del divisor de tensión de la NTC.

Figura17. Conexión de la NTC con Arduino DUE.

Figura18. Tabla de la relación resistencia- temperatura aportada por el fabricante.

Figura19. Gráfica de curva de calibración de la NTC.

Figura20. Gráfica de la recta de calibración de la NTC linealizada.

Figura21. Esquema de circuitos recomendados por el fabricante.

Figura22. Conexión del LM355 con el microprocesador Arduino DUE.

Figura23. Gráfica de la recta de calibración obtenida con el sensor LM355.

Figura24. Termopar tipo K.

Figura25. Esquema del puente de Wheatstone.

Figura26. Esquema interno del INA126.

Figura27. Conexionado con el microcontrolador.

Figura 28. Esquema final de las PT100 con el Arduino DUE.

Figura29. Pt100 recta de calibración del sensor resistivo.

Figura30. Pt100 salida amplificador operacional de instrumentación.

Figura31. Relación temperatura-resistencia.

Figura32. Conexionado de sensor de ultrasonidos con las placas Arduinos.

Figura33. Diagrama de tiempo del sensor de ultrasonidos HC-SR04.

Figura34. Conexionado de una electro-válvula con el Arduino DUE.

Figura 35. Esquema del circuito de una electro-válvula.

Figura36. Conexión de los sensores de nivel con la placa Arduino DUE.

Figura37. Conexión sensor de turbidez con módulo V1.0 a placa Arduino DUE.

Figura38. Tabla gráfica de los resultados obtenidos con el sensor de turbidez

Figura39. Conexionado de módulo HC-06 con Arduino UNO y Arduino DUE.

Figura40. Conexionado del circuito empleado para el LCD.

Figura 41. Fases del proceso de vinculación con módulo bluetooth HC-06.

Figura42. Descarga de aplicación final mediante código QR.

Figura43. Código QR disponible solo durante 2 horas después de la solicitud.

Figura44. Interfaz final de aplicación.

Figura45. Bloques de programación empleados para la app.

Figura46. Diagrama correspondiente al funcionamiento general de la pecera automatizada.

Figura47. Diagrama asociado a la pantalla LCD.

Figura48. Diagrama de los sensores de nivel y válvulas.

- Figura49. Diagrama del funcionamiento de los transductores NTCs.
- Figura50. Diagrama correspondiente a los sensores LM335.
- Figura51. Diagrama vinculado a los sensores PT100.
- Figura52. Diagrama asociado al sensor de ultrasonidos.
- Figura53. Diagrama asociado al sensor de turbidez.
- Figura54. Diagrama del módulo bluetooth HC-06.
- Figura55. Diagrama vinculado a la aplicación utilizada por el dispositivo móvil.
- Figura56. Diagrama de Gantt.
- Figura57. Módulo bluetooth HC-06.
- Figura58. Esquema interno de una electro-válvula.
- Figura59. Bomba de succión de agua.
- Figura60. Diagrama hardware.
- Figura61. Resultados obtenidos de los diferentes sensores de temperatura.
- Figura 62. Porcentaje de opacidad del nivel del agua de la pecera automatizada.
- Figura 63. Distancia obtenida por el sensor de ultrasonidos según incrementaba el nivel del agua.
- Figura 64. Estados de las electro-válvulas que se pueden dar según los sensores de nivel.
- Figura 65. Datos obtenidos con la aplicación móvil.
- Figura66. Esquema circuito electro-válvulas.
- Figura67 Layer SST.
- Figura68. Layer Bottom.
- Figura 69 Esquema circuito sensores de nivel.
- Figura70 Layer SST.
- Figura71. Layer Bottom.
- Figura72. Esquema circuito sensores NTC.
- Figura73 Layer SST.
- Figura74. Layer Bottom.

Figura 75. Esquema circuito sensores LM335.

Figura76 Layer SST.

Figura77. Layer Bottom.

Figura78. Esquema circuito sensores PT100.

Figura79 Layer SST.

Figura80. Layer Bottom.

- **LISTA DE TABLAS**

Tabla1: Resultados obtenidos con sensor NTC

Tabla2: Resultados obtenidos con sensor LM335

Tabla3: Resultados obtenidos con sensor PT100

Tabla4: Resultados obtenidos con sensor de turbidez

Tabla5: Conexiones módulo HC-06 con un micro controlador Arduino

Tabla 6: Diferentes fases del proyecto

Tabla7: Componentes iniciales

Tabla8: Componentes electrónicos

Tabla9: Material no electrónico.

1.2 CONTEXTO Y DIAGRAMAS DE BLOQUE

1.2.3 Contexto

Hoy en día, el uso de dispositivos electrónicos que poseen sistemas de medida de control son ya muy habituales. El objetivo principal en la mayoría de los casos, consiste en obtener información muy exacta sobre los procesos físicos del entorno. Debido a las limitaciones humanas, existen muchas variables que no se pueden detectar fácilmente.

Los sensores son dispositivos que ayudan en esta tarea. Estos son capaces de adquirir datos y convertirlos en señales eléctricas. Se trata de una forma de llevar la información desde la parte analógica, a la parte digital con el fin de que los datos puedan ser procesados. Dependiendo de su procedencia, algunas señales iniciales deberán ser acondicionadas previa y posteriormente al proceso

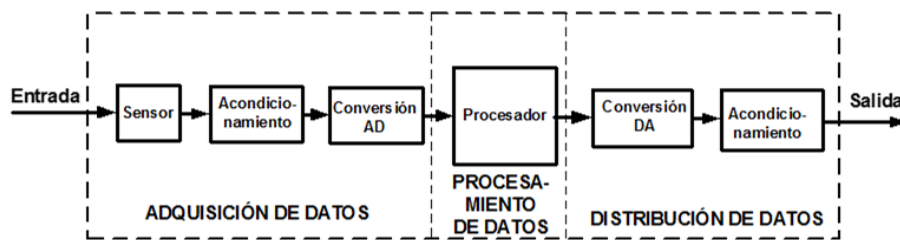


Fig1. Diagrama de un sistema de medida.

Este proyecto, consiste en el estudio y materialización de uno de estos sistemas de medida. A lo largo de este trabajo se detallará el seguimiento desde su inicio teórico, pasando por su diseño y finalizando con su montaje.

Se trata concretamente de una pecera automatizada, que posee sensores de temperatura y de nivel de agua, encargados de captar la información del entorno. También tiene un par de electro-válvulas que se abrirán o cerrarán dependiendo de estos sensores. Toda esta parte hardware recoge la información y la lleva hasta un micro-controlador, en este caso un Arduino DUE. Aquí entra en juego la parte software, la cual se encarga de que la información se procese correctamente y se envíe de vuelta.

La información podrá ser visible a través de una pantalla LDC o a través de un dispositivo móvil, mediante una aplicación android que se enlaza al Arduino a través de un módulo bluetooth. Otra de las funciones de esta información procesada, será controlar el autollenado de la pecera a través de las electro-válvulas. En la figura siguiente se puede ver el diagrama de bloques general de la pecera automatizada. Consiste en la unión de varios componentes hardware los cuales serán controlados por un microcontrolador Arduino DUE.

1.2.4 Diagrama de bloques

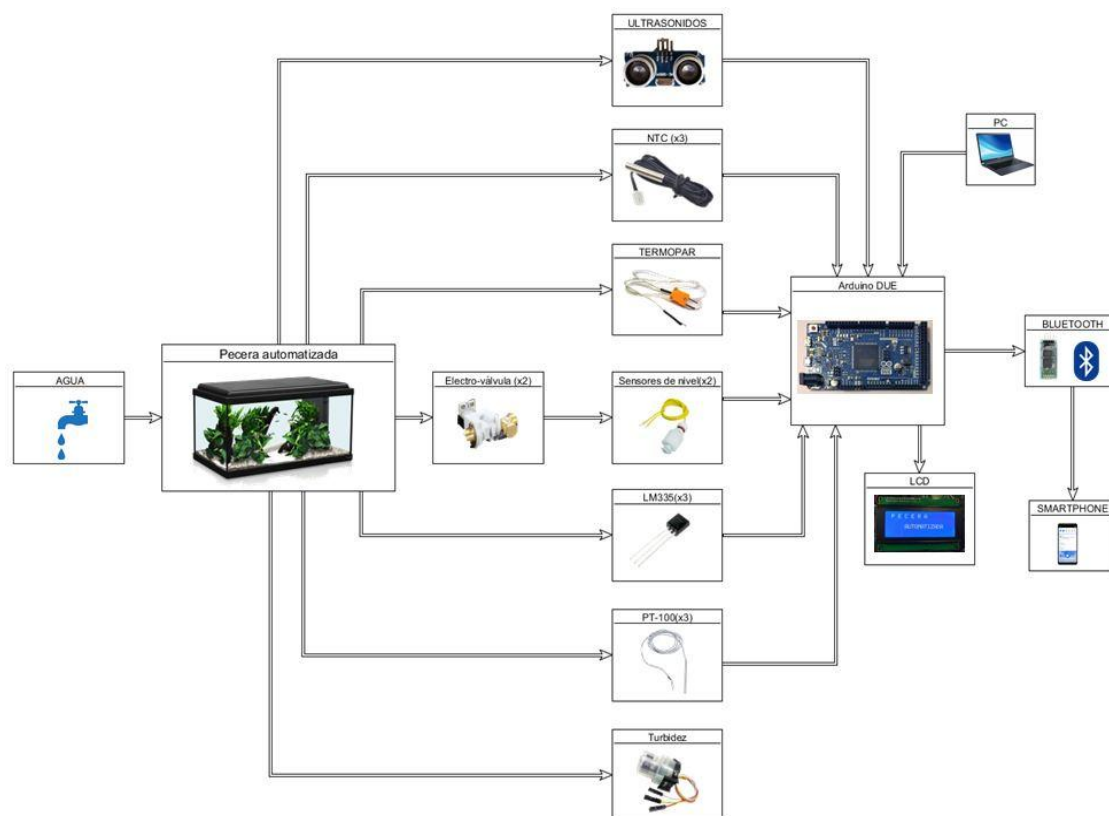


Fig2. Diagrama de bloques de la pecera automatizada.

1.3 OBJETIVOS Y ALCANCE DEL TRABAJO

El objetivo principal de este proyecto, consiste en implementar en una pecera una serie de dispositivos de medida y de control. Con los dispositivos de medida se quiere comparar los valores de temperatura obtenidos entre ellos y por otro lado con los de control poder conseguir el auto llenado de la pecera. Además, estos valores estarán disponibles a través de una pantalla LCD, la cual se podrá manipular con un par de pulsadores dependiendo de la información que se quiera ver en ese momento. Otra opción será válida con un dispositivo móvil enlazado vía bluetooth, el cual permitirá ver el seguimiento de los datos obtenidos con los sensores resistivos de temperatura LM335.

Con la realización de este proyecto también se persigue alcanzar una serie de objetivos generales:

- Aplicar y ampliar todos los conocimientos teóricos adquiridos en esta y otras asignaturas y llevarlos a un terreno práctico.
- Estudio y comparación de los distintos dispositivos electrónicos existentes en el mercado y selección de los más convenientes para este proyecto.
- Entendimiento y control del enlace entre software y hardware de los componentes implementados.
- Comprensión y conocimiento a un nivel superior sobre el microcontrolador Arduino DUE, así como de su entorno de programación y las posibles ventajas y desventajas de sus usos en este proyecto.
- Diseño, desarrollo y posterior montaje final de un prototipo en relación con las especificaciones iniciales.
- Afrontar posibles problemas que puedan surgir a lo largo del montaje y aplicar posibles medidas para lograr su resolución.

1.4 DESCRIPCIÓN DE TAREAS Y PROCEDIMIENTOS

Para poder llevar a cabo este proyecto, ha sido necesario pasar por tres fases diferentes. La primera, ha sido un estudio teórico sobre el objeto en el que se situará este sistema de medida. Es decir, la forma física en la que se representará, en este caso una pecera automatizada.

La segunda fase lo forman la parte hardware, la parte software y los diferentes circuitos necesarios y sus posteriores cálculos para conseguir un funcionamiento correcto.

Y finalmente la parte del montaje, la cual ha ocupado un mayor tiempo debido a los problemas surgidos según se iban realizando los montajes de los circuitos. Una vez finalizadas estas tres fases se ha podido proceder a la creación de una aplicación Android, gracias al programa MIT App Inventor. Ésta es necesaria para poder acceder a los datos en tiempo real de los grados de temperatura obtenidos con las tres LM335s que forman parte de esta pecera.

1.5 IMPLEMENTACIÓN HARDWARE

A continuación, se describen todos los componentes físicos elegidos y que forman parte de esta pecera automatizada. Mediante estos dispositivos, se llevan a cabo las tareas principales, ya que ayudan tanto en la recogida de información como en el procesamiento de esta.

Se describe cada uno de estos elementos, así como la tarea que desempeñan en la pecera automatizada. A continuación, se hace una breve descripción acerca del microprocesador elegido y sus diferentes opciones. Finalmente se termina con los cálculos realizados, necesarios para integrar cada dispositivo a la pecera.

1.5.1 COMPONENTES

1.5.1.1 NTC

Los termistores NTC, son sensores resistivos de coeficiente de temperatura negativo, que se fabrican con óxidos metálicos. Su funcionamiento, se basa en la variación de la resistividad que presenta un semiconductor con la temperatura. Esto es, al aumentar la temperatura, disminuye su resistencia.

En la pecera automatizada, se ha utilizado tres sensores de este tipo. El objetivo es realizar comprobaciones de los resultados obtenidos con cada uno de ellos. A su vez estos van a formar parte de una serie de sensores de varios tipos, los cuales se encontrarán alrededor de la pecera.

Debido a la falta de tiendas cercanas para su compra y a la comparación de precios, finalmente se han obtenido estos productos a través de una plataforma de compra. Al realizar la compra también se ha tenido en cuenta su no linealidad, por lo que se han adquirido componentes extras, necesarios para realizar un circuito de acondicionamiento con un divisor de tensión.



Fig3. Sensor NTC utilizado en el proyecto.

1.5.1.2 LM335

Entre los sensores elegidos para este proyecto, se encuentran los sensores LM335. Al igual que los NTCs, se emplean tres sensores de este tipo. Se han elegido debido a su alta precisión y fácil calibración. Funcionan como un Zener de dos terminales. Tiene un voltaje de ruptura directamente proporcional a la temperatura absoluta a $10\text{mV}/^\circ\text{K}$, con menos de 1Ω de impedancia dinámica. Cuando es calibrado a 25°C , el LM335 tiene típicamente menos de 1°C de error sobre 100°C de rango de temperatura.

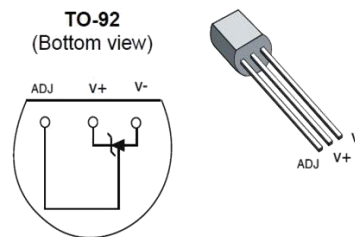


Fig4. Encapsulado TO-92 LM335.

Otro de los objetivos que se ha querido realizar con este dispositivo, es que además de la temperatura del aire, nos proporcione la temperatura del agua. Por este motivo se ha realizado un encapsulado improvisado con un material aislante.

1.5.1.3 TERMOPAR-TIPO K

Otro tipo de los sensores empleados para realizar la posterior comparación de los grados obtenidos, son los termopares. Concretamente se ha empleado un termopar tipo K, de doble canal. Estos suelen tener un rango de funcionamiento de -20 a 260°C . Su elección se ha debido a la facilidad de trabajar con ellos, ya que al conectarse directamente en el multímetro, no necesitan ningún acondicionamiento previo. En esta ocasión solamente se emplea un termopar, que estará cerca de los demás sensores.



Fig5. Termopar tipo K.

1.5.1.4 RTD-PT100

El último tipo de sensor que compone la pecera automatizada, es el PT100 y se trata de un tipo de sensor RTD (Resistor Temperature Detector). Los RTDs son sensores resistivos, cuyo funcionamiento se basa en la variación de la resistividad, que presenta un conductor con la temperatura. Se denomina PT100 porque a 0°C la resistencia que podemos medir entre sus terminales es de 100 Ω . Por lo general, la variación es bastante lineal en márgenes amplios de temperatura y posee un coeficiente de temperatura positivo.



Fig6. Sonda Pt100.

Al igual que casi todos los sensores de temperatura elegidos, la pecera consta de tres sensores de este tipo. La elección de este sensor se ha realizado principalmente por la labor que conlleva su acondicionamiento y preparación para poder obtener una señal óptima.

Esto es debido, a que a su salida se obtendrá una señal muy pequeña. Por ello, para su acondicionamiento se ha utilizado un puente de wheatstone, seguido de un amplificador operacional de instrumentación, lo cual se explicará detalladamente más adelante en el apartado cálculos.

1.5.1.5 SENSOR DE ULTRASONIDOS PIEZOELÉCTRICO HC-SR04

Otro de los componentes hardware que forman parte de la pecera automatizada, es el sensor de ultrasonidos, cuya función principal consistirá en medir la distancia a la que se encuentra el agua en cada momento, desde la base de la pecera, hasta el tope donde se va a encontrar el sensor.

Está compuesto por dos transductores piezoeléctricos. Se trata de un transmisor que emite un pulso de ultrasonidos, el cual rebota sobre un determinado objeto tanto sólido como líquido, en este caso el agua y un receptor que detecta la reflexión provocada por el pulso.

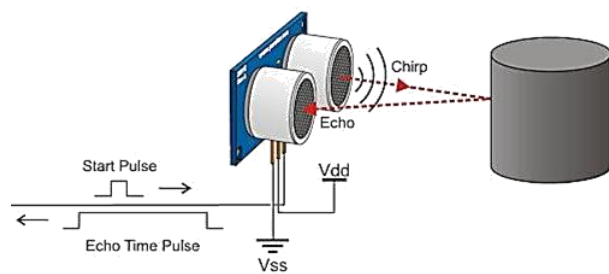


Fig7. Funcionamiento modo eco del sensor de ultrasonidos.

1.5.1.6 ELECTRO-VÁLVULAS

Dos dispositivos fundamentales en esta pecera, van a ser los dos electro-válvulas. Estas van a estar en esquinas opuestas de la pecera y van a actuar como compuertas, que van a permitir la entrada y salida del agua, en función de los sensores de nivel. Conectadas a cada una de ellas habrá una manguera, así una tendrá acceso a un grifo de agua, mientras que la otra estará cerca de un desagüe.

La electro-válvula es un dispositivo mecánico con el cual se puede iniciar o detener la circulación de líquidos o gases. Esto se realiza gracias a una pieza movible denominada émbolo, la cual comunica el orificio de entrada y salida con su movimiento realizado por acción magnética mediante una bobina solenoide.

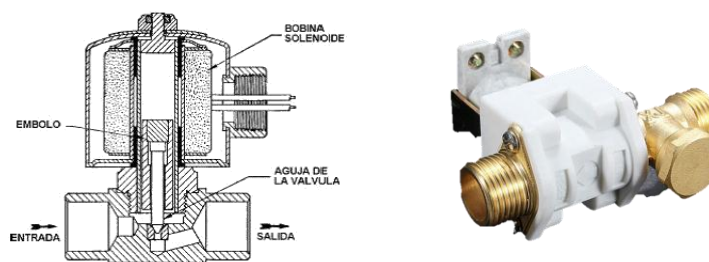


Fig8. Interior y exterior de una electro-válvula.

1.5.1.7 SENSOR DE NIVEL DE AGUA

Los sensores de nivel van a estar relacionados estrechamente con las electro-válvulas, ya que en función de estos las válvulas realizarán una acción de apertura

o cierre. La pecera tiene dos sensores de nivel situados a diferente altura, de forma que delimiten el llenado máximo o mínimo posible.

Su funcionamiento interno trata de un contacto abierto, el cual, si el nivel de agua está por debajo de la sonda, el contacto permanece abierto y por lo tanto no circula corriente (GND para el arduino). Por el contrario, si el nivel del agua está a la altura de la sonda o la sobrepasa, el contacto se cierra y por lo tanto circula corriente (Vcc para el arduino).



Fig9. Sensor de nivel.

1.5.1.8 SENSOR DE TURBIDEZ

Se ha utilizado el sensor DF Robot SKU: SEN0189. La forma de funcionamiento, consiste en detectar la proporción de sólidos suspendidos en el agua al medir la transmisión y la tasa de dispersión de la luz emitida. Además, viene con un módulo driver de conexión V1.0, por lo que no requiere de un acondicionamiento previo. Este módulo permite elegir la opción de uso entre analógico y digital, a través de un interruptor y su diseño se basa en el amplificador operacional dual LMV3558. Mientras que la opción digital es un circuito comparador, la opción analógica es un circuito búfer.

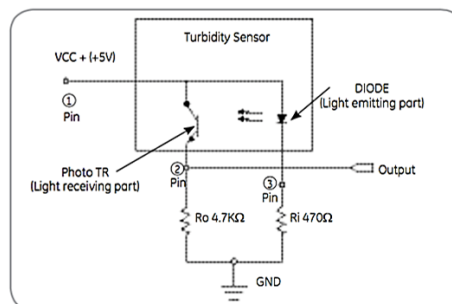


Fig10. Esquema interno del sensor de turbidez.



Fig11. Sensor de turbidez con y módulo V1.0 de conexión.

1.5.1.9 MÓDULO BLUETOOTH HC-06

El módulo bluetooth HC-06 que forma parte de la pecera, será el que deje acceder al dispositivo móvil a los datos obtenidos por el microprocesado Arduino, necesarios para poder interactuar con la aplicación móvil. Esto será posible gracias a que enlazará con el Smartphone mediante bluetooth.

Para la elección se planteó entre dos dispositivos el módulo HC-06 (esclavo) y el módulo HC-05 (maestro/esclavo). La diferencia entre maestro y esclavo es que en modo esclavo es el dispositivo quien se conecta al módulo, mientras que en modo maestro, es el módulo quien se conecta con un dispositivo. Físicamente se puede ver en la imagen que el módulo HC-05 posee un pin más.

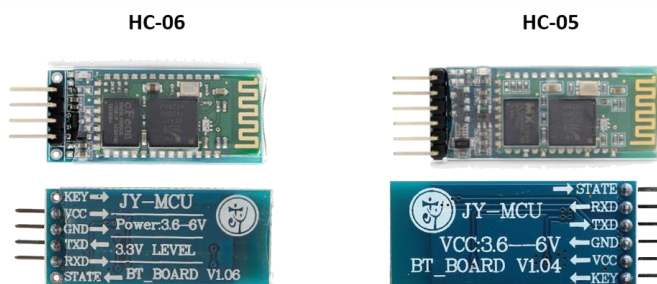


Fig12. Módulos bluetooth HC-06 (esclavo) y HC-05 (maestro/esclavo).

1.5.1.10 PANTALLA LCD 20X4 CON MÓDULO I2C

Para poder hacer visible todos los datos obtenidos en el proceso es necesario contar con una pantalla o LCD. La principal diferencia entre las opciones a elegir, se diferencian principalmente en el tamaño y en la conexión. La elección final ha sido la de un LCD de 20x4, ya que de esta forma se podría mostrar más información de una sola vez. Además consta de un módulo I2C en la que un micro actúa como maestro y se comunica con el resto de periféricos como esclavos

compartiendo las conexiones marcadas como SDA y SCL. El maestro manda el dato por el bus y la dirección del periférico al que va destinado. Por eso, cada periférico solo puede tener una dirección única para evitar el conflicto.

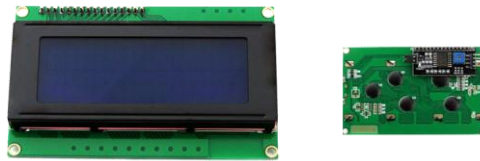


Fig13. Pantalla lcd 20x4 con módulo I2C.

1.5.2 SISTEMA DE PROCESAMIENTO

Una vez seleccionados los elementos de hardware necesarios para poder realizar las tareas, se debe elegir un sistema de procesamiento.

Dada la naturaleza y tamaño de nuestro proyecto, se ha elegido un micro-controlador de la gama Arduino. Otra opción válida habría sido la de un microprocesador. Ambos poseen sus ventajas y desventajas, pero debido a que el micro-controlador presenta una velocidad de operaciones y potencia de cálculo mayor frente al microprocesador, se ha seleccionado a éste como mejor opción. Dentro de la gama de micro-controladores Arduino se han utilizado principalmente dos modelos. Por un lado, se ha empleado el Arduino UNO para la realización de pruebas aislada de algunos dispositivos, mientras que el Arduino DUE se ha empleado como micro-controlador final para formar parte del proyecto.

Esta elección se ha realizado por la mayor potencia y número de pines de entrada/salida que presenta este segundo tipo.

1.5.2.1 MICROCONTROLADOR ARDUINO

Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware y software libre, que incorpora un microcontrolador ATMEL re-programable. Se trata de una placa PCB (Printed Circuit Board /Placa de Circuito Impreso), que implementa un determinado diseño de circuitería interna.

Consta de una interfaz de entrada, que es una conexión en la que podemos conectar en la placa diferentes tipos de periféricos. La información de estos periféricos se trasladará al microcontrolador y este se encargará de procesar los datos que le lleguen a través de ellos. También cuenta con una interfaz de salida, que es la que se encargará de llevar la información que se ha procesado de vuelta a los periféricos de salida.

1.5.2.2 TIPOS DE ARDUINO

Dentro del proyecto Arduino, se han fabricado diferentes modelos de placas Arduino oficiales, las cuales comparten su diseño básico. Cada una, se especializa en un propósito diferente (Internet, impresión 3D, proyectos sencillos) y posee características que las diferencian como el tamaño físico, número de pines E/S, modelo del microcontrolador, precio... etc).

A continuación podemos ver algunas de las placas Arduino más usadas en el mercado.



Fig14. Distintos tipos de Arduino.

Sin duda el modelo más usado para proyectos y extendido es el Arduino UNO, esto es debido a su precio y a la facilidad de poder realizar proyectos sencillos para gente con pocos conocimientos de electrónica, como para proyectos más complejos para gente más especializada en el tema.. A continuación se puede ver una imagen de los dos micro-controladores empleados en este proyecto.

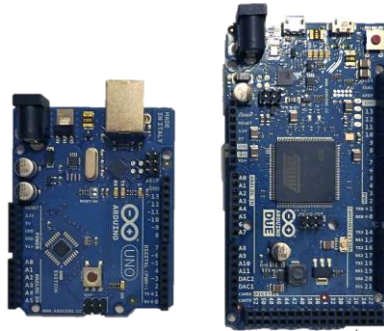


Fig15. Placa Arduino UNO y DUE.

1.5.3 CÁLCULOS HARDWARE

En este apartado se va a comentar el conexionado que se ha llevado a cabo, para relacionar cada elemento hardware con el microprocesador Arduino DUE. También se detallarán los cálculos imprescindibles para poder acondicionar previamente cada sensor en caso de necesitarlo, antes de la conexión final.

La mayoría de estos cálculos, están relacionados con el acondicionamiento de la señal de entrada, debido a que es una señal muy pequeña o porque son dispositivos no lineales. En otros casos, simplemente necesitan estar en un circuito determinado para poder llevar a cabo la tarea designada.

1.5.3.1 NTC

Los termistores NTC, son sensores resistivos de coeficiente de temperatura negativo que necesitan ser acondicionados previamente a ser usados. Para poder llevar a cabo este acondicionamiento para conectarlo al microcontrolador, hay varias opciones, como por ejemplo el uso de una fuente corriente, o como la utilizada en este caso para la pecera automatizada, con un divisor de tensión.

Se tiene a favor que la señal que se obtendrá posteriormente (V01), no resultará demasiado pequeña, por lo que no hará falta tener que amplificarla.

A continuación se van a describir los cálculos necesarios para su acondicionamiento, para ello se necesitará calcular el valor de su resistencia y temperatura final. Estos cálculos también se utilizarán para realizar la programación en Arduino de estos sensores en apartados posteriores. Después se realizarán los cálculos para la linealización del dispositivo.

Para el divisor de tensión se ha utilizado un potenciómetro de 10K Ω , como se puede ver en la figura a continuación.

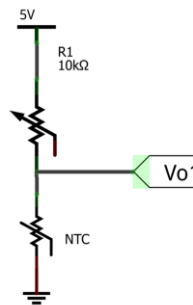


Fig16. Esquema del divisor de tensión de la NTC.

En la siguiente figura podemos ver como ha quedado finalmente el conexionado del termistor con el microcontrolador, los pines de las entradas analógicas que se usarán son A0,A1,A2.

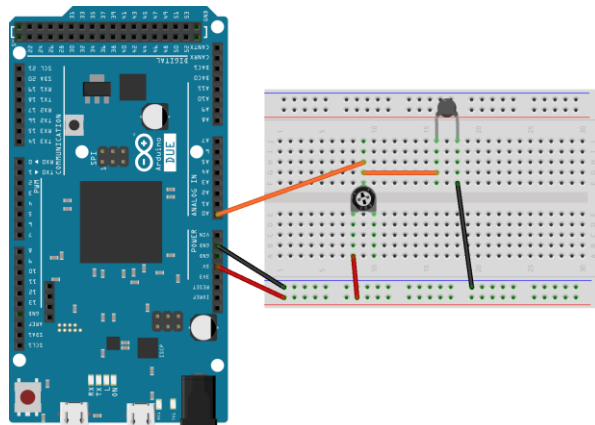


Fig17. Conexionado de la NTC con Arduino DUE.

Para poder obtener la temperatura final, se necesitará calcular el valor de la resistencia de la NTC y por lo tanto despejar ésta, utilizando el valor de la tensión de salida que se obtiene del divisor de tensión. Una vez calculada la tensión V01 obtenida por nuestro sensor, y sabiendo los valores de VCC(5V) y R1(10KΩ).

Valor de resistencia a la temperatura media:

$$V01 = \frac{R_{ntc} * V_{cc}}{(R_{ntc} + R1)}$$

$$(R_{ntc} + R1) = \frac{R_{ntc} * V_{cc}}{V01}$$

$$\frac{R1}{R_{ntc}} = \frac{V_{cc}}{V01} - 1$$

$$R_{ntc} = \frac{R1}{\frac{V_{cc} - V01}{V01}}$$

Valor de la temperatura

Una vez obtenido el valor de la Rntc, mediante su fórmula obtenemos el valor de la temperatura. Como se sabe esta fórmula proviene del modelo matemático que relaciona la resistencia y temperatura del dispositivo, la cual se puede apreciar con una forma exponencial en la figura siguiente:

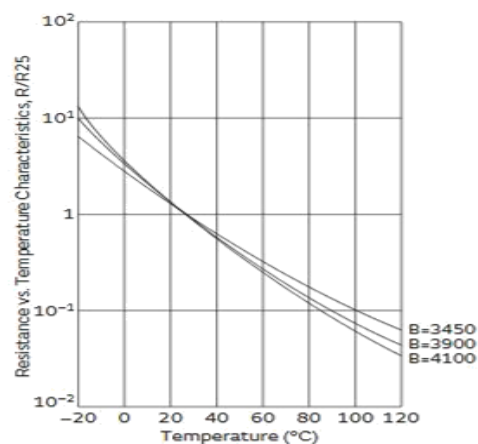


Fig18. Tabla de la relación resistencia- temperatura aportada por el fabricante.

$$\begin{aligned}
 R_{T_m} &= R_0 e^{\beta \left(\frac{1}{T_m} - \frac{1}{T_0} \right)} \\
 \ln \left(\frac{R_{T_m}}{R_0} \right) &= \ln * e^{\beta \left(\frac{1}{T_m} - \frac{1}{T_0} \right)} \\
 \ln \left(\frac{R_{T_m}}{R_0} \right) &= \beta * \left(\frac{1}{T_m} - \frac{1}{T_0} \right) \\
 \ln \left(\frac{R_{T_m}}{R_0} \right) &= \beta * \left(\frac{1}{T_m} - \frac{1}{T_0} \right) \\
 \ln \left(\frac{R_{T_m}}{R_0} \right) + \frac{\beta}{T_0} &= \left(\frac{\beta}{T_m} \right) \\
 T_m &= \left(\frac{\beta}{\ln \left(\frac{R_{T_m}}{R_0} \right) + \frac{\beta}{T_0}} \right)
 \end{aligned}$$

Como se ha comentado anteriormente la NTC necesita ser linealizada. Entre las opciones para hacerlo, pueden ser mediante una resistencia en paralelo, o a través de un divisor resistivo, el cual ha sido el método empleado en el caso de la pecera.

Con la primera opción, aunque la resistencia resultante no sea totalmente lineal, su variación con la temperatura es menor, es decir se eleva la linealidad a costa de reducir la sensibilidad. Para esta pecera automatizada finalmente se ha utilizado este segundo método, un divisor resistivo. Se debe saber que en este caso, no se puede utilizar cualquier resistencia, hay que especificar el área de la curva de la NTC con la que se va a trabajar. El rango de temperatura elegido, estará comprendido entre 10°C y 30°C.

Linealización de la NTC

Temperatura media: $(10+30)/2 = 20 \text{ }^{\circ}\text{C}$

Valor de resistencia a la temperatura media:

$$\begin{aligned}
 R_{T_m} &= R_0 e^{\beta \left(\frac{1}{T_m} - \frac{1}{T_0} \right)} \\
 R_{T_m} &= 10K * e^{3950 \left(\frac{1}{20+273} - \frac{1}{25+273} \right)} \\
 R_{T_m} &= 12538.20 \Omega
 \end{aligned}$$

$$R_L = \frac{\beta - 2T_M}{\beta + 2T_M} R_{T_m}$$

$$R_L = \frac{3950 - 2(20 + 273)}{3950 + 2(20 + 273)} 12538.20 = 9K3 \Omega$$

A continuación se adjunta la tabla de valores obtenidos con estos resultados anteriores y dos gráficas de la NTC sin linealizar y linealizada.

NTC			
Temp(°C)	Rntc(Ohm)	RL(Ohm)	Vo(V)
10	20189,22	9299	3,27
11	19220,98	9299	3,22
12	18305,48	9299	3,17
13	17439,55	9299	3,12
14	16620,18	9299	3,07
15	15844,61	9299	3,01
16	15110,22	9299	2,96
17	14414,59	9299	2,91
18	13755,44	9299	2,85
19	13130,64	9299	2,80
20	12538,20	9299	2,74
21	11976,25	9299	2,69
22	11443,04	9299	2,64
23	10936,94	9299	2,58
24	10456,41	9299	2,53
25	10000,00	9299	2,48
26	9566,37	9299	2,42
27	9154,25	9299	2,37
28	8762,45	9299	2,32
29	8389,84	9299	2,27
30	8035,39	9299	2,22

Tabla 1.Resultados obtenidos con sensor NTC.

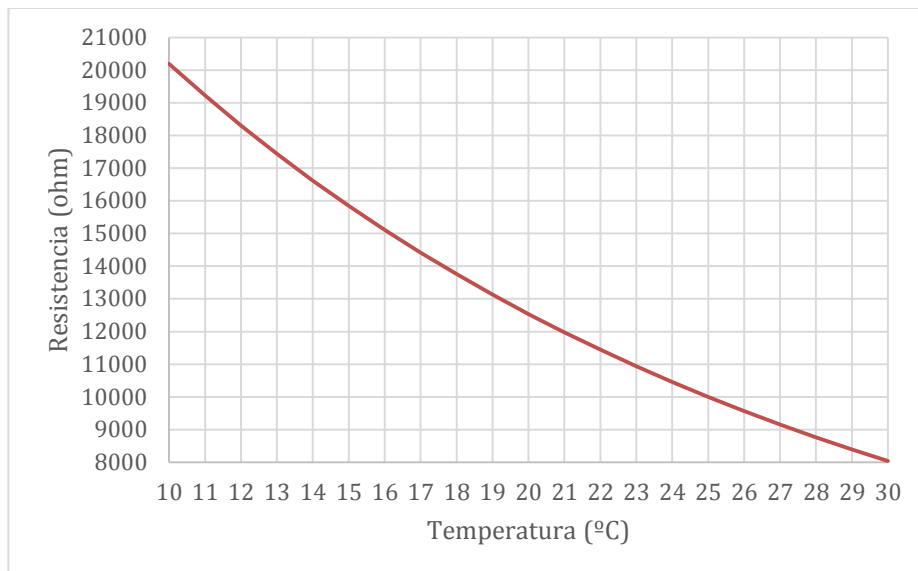


Fig19. Gráfica de curva de calibración de la NTC.

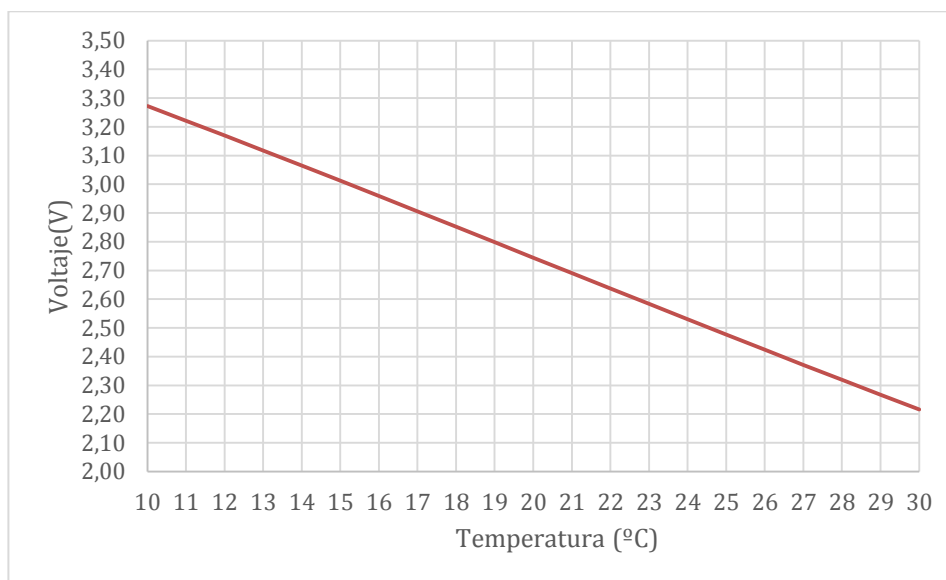


Fig20. Gráfica de la recta de calibración de la NTC linealizada.

1.5.3.2 LM335

El sensor de temperatura LM335 como se ha comentado anteriormente, tiene una gran ventaja y es que no necesita ser linealizado. Esto es debido a que funciona de forma similar a un zener y se comporta con una sensibilidad de 10mV/K . Está calibrado en grados Kelvin y trabaja en un rango que va desde $233,15^\circ\text{K}$ a $373,15^\circ\text{K}$ (-40°C a $+100^\circ\text{C}$).

A pesar de no necesitar ser linealizado, este sensor no puede ser conectado directamente al Arduino DUE. Para poder trabajar con él, es necesario saber sus características:

- Exactitud inicial de 1°C ($\pm 0.5^\circ\text{C}$ con calibración)
- Salida de $10\text{mV}/^\circ\text{K}$
- Impedancia dinámica menor a 1Ω
- Polarizado por corriente. Opera desde $450\mu\text{A}$ a 5mA
- Encapsulado: TO-92

Además, en la hoja de características se informa que para trabajar con una lectura mucho más precisa, se recomienda la utilización del tercer pin (adj) conectado a un potenciómetro de $10\text{K}\Omega$.

En la figura siguiente se pueden apreciar los dos circuitos que se recomienda en su hoja de características. La primera figura hace referencia a un circuito básico, y la segunda a un circuito calibrado de forma más precisa

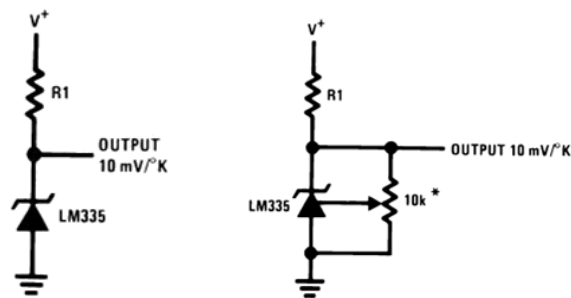


Fig21. Esquema de circuitos recomendados por el fabricante.

En esta otra figura se puede ver el conexionado final con el Arduino DUE, el cual se ha realizado utilizando dos potenciómetros de $10\text{K}\Omega$. Uno de los potenciómetros

Esta “calibración” una vez hecho el circuito consiste en modificar el potenciómetro de ajuste del pin (ADJ) hasta conseguir un valor de 2.983V como V_{out} , tal como se indica en la hoja de características. Esta es la tensión que debería salir suponiendo una temperatura ambiente de 25°C.

$$V_{out} = 0,01 * (T^{\circ}C + 273) \quad (2)$$

30

LM335			
<i>Temp(°C)</i>	<i>Temp(K)</i>	<i>S(mV/°C)</i>	<i>Vo(V)</i>
10	283	10	2,83
11	284	10	2,84
12	285	10	2,85
13	286	10	2,86
14	287	10	2,87
15	288	10	2,88
16	289	10	2,89
17	290	10	2,90
18	291	10	2,91
19	292	10	2,92
20	293	10	2,93
21	294	10	2,94
22	295	10	2,95
23	296	10	2,96
24	297	10	2,97
25	298	10	2,98
26	299	10	2,99
27	300	10	3,00
28	301	10	3,01
29	302	10	3,02
30	303	10	3,03

Tabla 2 .Resultados obtenidos con sensor LM335.

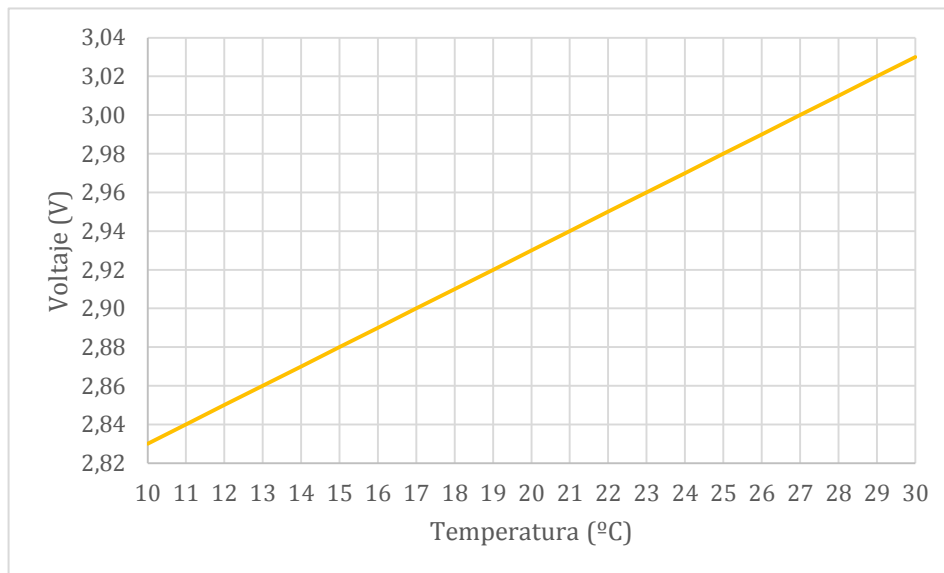


Fig23. Gráfica de la recta de calibración obtenida con el sensor LM355.

1.5.3.3 TERMOPAR-TIPO K

El termopar tipo K no requiere de ningún circuito especial para el acondicionamiento de la señal, ya que se trata de un sensor de doble canal que se conecta directamente con un multímetro. Se ha elegido este sensor, pensando más en una forma de obtener datos precisos para poder posteriormente compararlos con los demás resultados.



Fig24. Termopar tipo K.

1.5.3.4 RTD-PT100

La PT100, es un sensor RTD (*Resistor Temperature Detector*) cuyo nombre viene dado debido a que a 0°C la resistencia que podemos medir entre sus terminales es de 100 Ω . Sin duda alguna es el modelo de RTD más usado en el mercado por sus características, principalmente el de su linealidad.

En el caso de la pecera, se han empleado tres sensores de este tipo como en muchos de los otros sensores elegidos, pero se describirá la relación sobre uno de ellos. A pesar de ser lineal, este sensor tiene un inconveniente y es que la señal que obtenemos de este sensor es muy pequeña. Por ello parte de su acondicionamiento va a consistir en la utilización de un amplificador operacional posterior a las señales que se obtendrán con el de Wheatstone. El amplificador operacional elegido es el INA126.

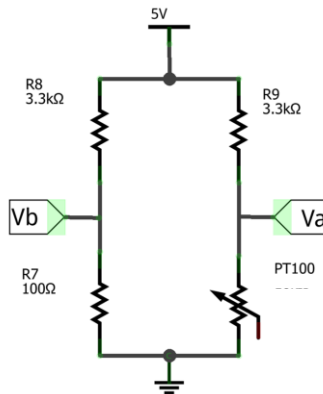


Fig25. Esquema del puente de Wheatstone

a

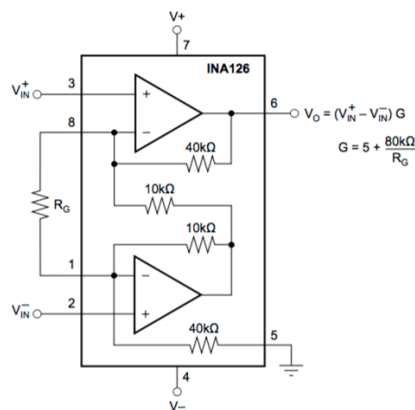


Fig26. Esquema interno del INA126.

Cabe destacar que en el puente de Wheatstone las dos resistencias de arriba y de abajo deben ser iguales. Y debido a que $100\ \Omega$ es el valor que arroja la PT100 a 0°C , la R7 tiene este valor. A continuación en las siguientes dos figuras se pueden ver por una lado el conexionado final realizado para conseguir el acondicionamiento de una de las PT100 con el microcontrolador Arduino DUE. Y en la siguiente figura podemos ver el esquema final con las 3 sensores.

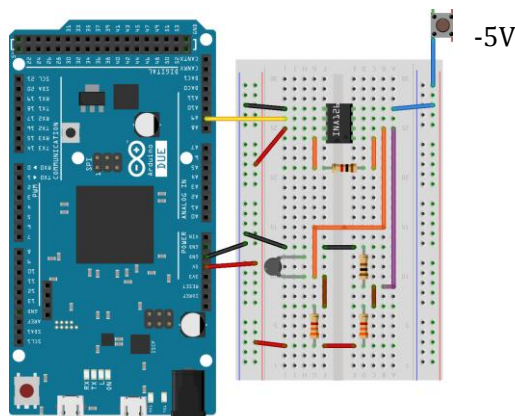


Fig27. Conexionado con el microcontrolador.

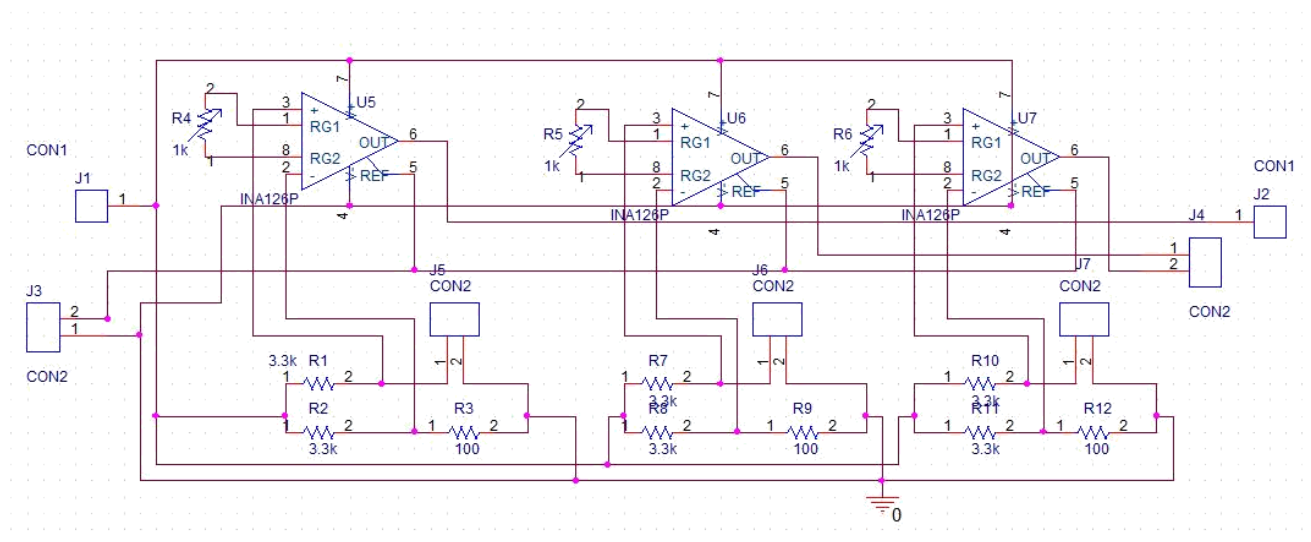


Fig28. Esquema final de las PT100 con el Arduino DUE.

Calculo de la resistencia a la temperatura (T):

$$R = R_0 \times (1 + \alpha \Delta T)$$

$$R = 100 \times (1 + (0.00358 \times \Delta T))$$

$$R (10^\circ\text{C}) = 103.85 \text{ Ohm}$$

$$R (20^\circ\text{C}) = 107.7 \text{ Ohm}$$

$$R (30^\circ\text{C}) = 111.55 \text{ Ohm}$$

A continuación se van a describir los cálculos necesarios para poder llevar a cabo la programación en Arduino, una vez calculada la tensión amplificada del sensor deberemos calcular la V^+ , la R_{pt100} y por último la temperatura obtenida. Tanto $V^+(V_a)$ como $V^-(V_b)$ la hemos sacado del puente de Wheatstone.

$$V_o = (V^+ - V^-)G$$

$$\frac{V_o}{G} = (V^+ - V^-)$$

$$V^+ = \frac{V_o + (V^- \times G)}{G}$$

$$V^+ = \frac{R_{pt100}}{R_3 + R_{pt100}} V_{cc}$$

$$\frac{V^+}{V_{cc}} = \frac{R_{pt100}}{(R_3 + R_{pt100})}$$

$$\frac{V^+}{V_{cc}} = \frac{R_{pt100}}{(R_3 + R_{pt100})}$$

$$(V^+ \times R_3) = R_{pt100} \times (V_{cc} - V^+)$$

$$R_{pt100} = \frac{(V^+ \times R_3)}{(V_{cc} - V^+)}$$

$$R = R_0 \times (1 + \alpha \Delta T)$$

$$\frac{R}{R_0} - 1 = \alpha \Delta T$$

$$\Delta T = \frac{\left(\frac{R}{R_0} - 1\right)}{\alpha}$$

Cálculos a la salida del INA126:

$$V_o = (V^+ - V^-)G$$

$$G = \left(\frac{80k}{R_g} + 5 \right) \Rightarrow R_g(G = 100) = 842.10 \text{ Ohm}$$

$$V^+ = \frac{R_{pt100}}{R_3 + R_{pt100}} V_{cc}$$

$$V^- = \frac{R_2}{R_1 + R_2} V_{cc} = \frac{100}{3200 + 100} 5 = 151.15mV$$

Resultados:

Resistencia(ohm)	RL(Ohm)	Vo(V)	INA126(mV)	INA126(V)
103,85	3200	0,15717	565,0	0,5650
104,24	3200	0,15773	621,4	0,6214
104,62	3200	0,15829	677,8	0,6778
105,01	3200	0,15886	734,2	0,7342
105,39	3200	0,15942	790,6	0,7906
105,78	3200	0,15999	847,0	0,8470
106,16	3200	0,16055	903,4	0,9034
106,55	3200	0,16111	959,7	0,9597
106,93	3200	0,16168	1016,0	1,0160
107,32	3200	0,16224	1072,4	1,0724
107,70	3200	0,16280	1128,7	1,1287
108,09	3200	0,16336	1185,0	1,1850
108,47	3200	0,16393	1241,3	1,2413
108,86	3200	0,16449	1297,5	1,2975
109,24	3200	0,16505	1353,8	1,3538
109,63	3200	0,16562	1410,0	1,4100
110,01	3200	0,16618	1466,3	1,4663
110,40	3200	0,16674	1522,5	1,5225
110,78	3200	0,16730	1578,7	1,5787
111,17	3200	0,16786	1634,9	1,6349
111,55	3200	0,16843	1691,1	1,6911

Tabla 3. Resultados obtenidos con sensor PT100.

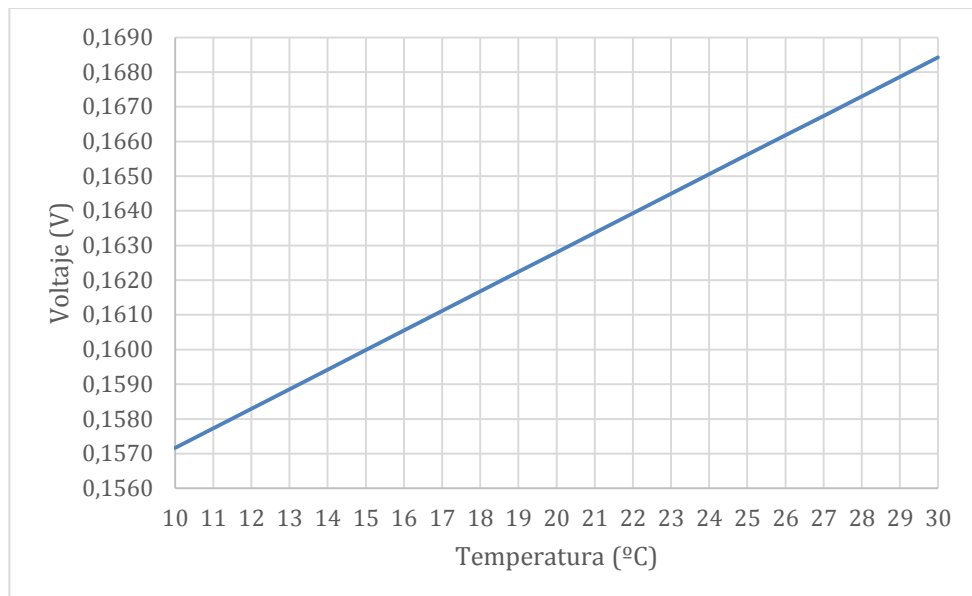


Fig29. Pt100 recta de calibración del sensor resistivo.

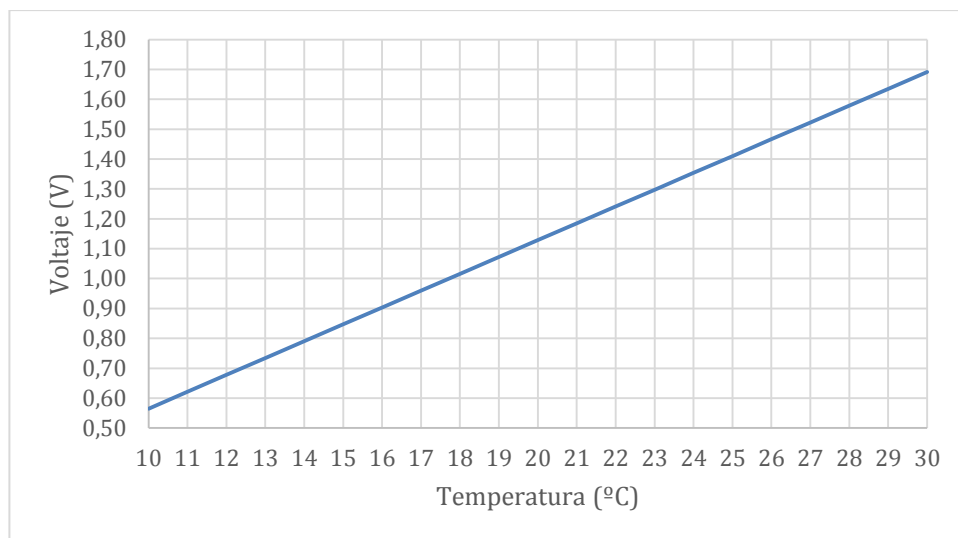


Fig30. Pt100 salida amplificador operacional de instrumentación.

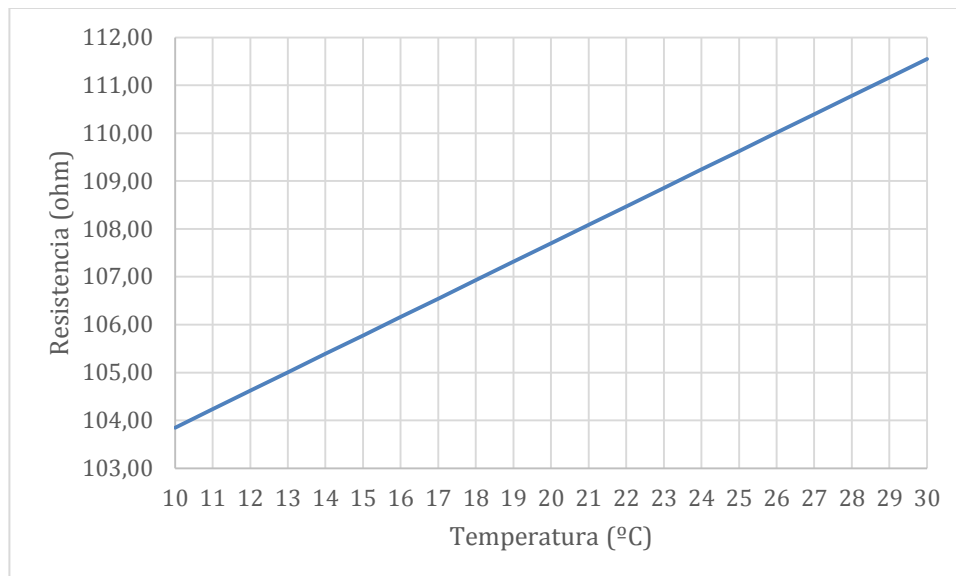


Fig31.Relación temperatura-resistencia.

1.5.3.5 SENSOR DE ULTRASONIDOS PIEZOELÉCTRICO HC-SR04

Para el uso del sensor de ultrasonidos HC-SR04, se han empleado dos placas Arduino. Como en muchos de estos sensores, se ha realizado una primera prueba aislada con la placa Arduino UNO, y posteriormente con Arduino DUE. En la figura siguiente se puede apreciar la conexión con Arduino UNO y DUE, la cual no ha tenido gran diferencia. Cabe destacar que los pines Trigger y Echo se han de conectar a las entradas/salidas digitales. El pinTrigger va a funcionar en modo salida y el pin Echo en modo entrada.

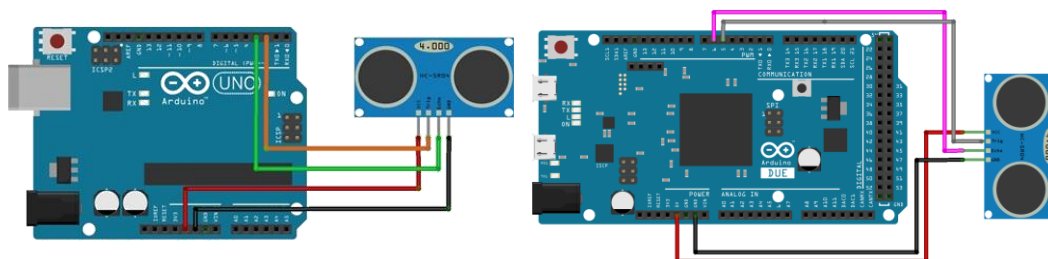


Fig32. Conexionado de sensor de ultrasonidos con las placas Arduinos.

El funcionamiento de sus pines viene marcado por el diagrama de tiempo de la siguiente figura.

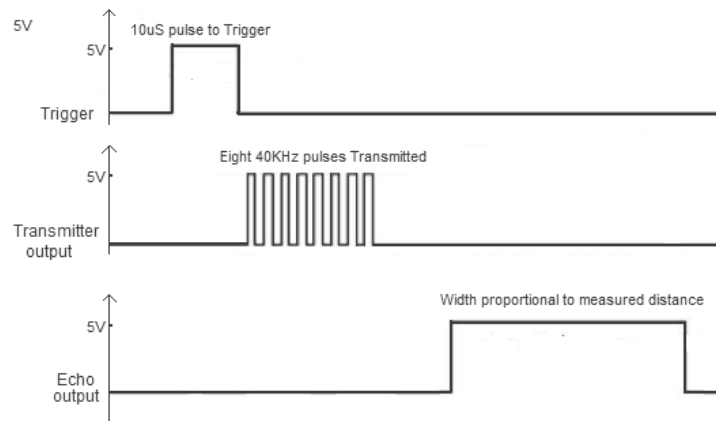


Fig33. Diagrama de tiempo del sensor de ultrasonidos HC-SR04.

Para que el módulo inicie su trabajo, es necesario enviar un pulso TTL de 5V con una duración de 10 µs a través del pin TRIG. Después, el módulo enviará ocho pulsos ultrasónicos de 40 kHz. Finalmente a través del pin ECHO, obtendremos la distancia en forma de pulsos con nivel de amplitud más baja. Los valores de la distancia estarán relacionados con el ancho de pulso.

Para el cálculo de la distancia, el sensor relaciona el tiempo entre el envío y la recepción de un pulso sonoro mediante una fórmula. Además, es necesario saber que la velocidad del sonido es aproximadamente de 343 m/s en condiciones de temperatura 20 °C, 50% de humedad y presión atmosférica a nivel del mar.

Transformando las unidades de m/s a cm/µs, obtenemos que el sonido tarda 29,2 µs en recorrer un centímetro y teniendo en cuenta que la distancia que recorre el pulso será de ida y vuelta dividimos por 2:

$$343 \frac{m}{s} * 100 \frac{cm}{m} * \frac{1}{1 * 10^6} \frac{s}{\mu s} = \frac{1}{29,2} \frac{cm}{\mu s}$$

$$Distancia(cm) = Velocidad(m/s) * Tiempo(\mu s)$$

$$Distancia(cm) = \frac{Tiempo(\mu s)}{(29,2 * 2)}$$

1.5.3.6 ELECTRO-VÁLVULAS

La pecera automatizada tiene dos electro- válvulas. Estas van a permitir el paso o salida del flujo del agua según sean accionadas o no por el Arduino. Las válvulas por las características que poseen deberán ser alimentadas a 12V por una fuente de tensión externa. Por protección del arduino frente a las contra corrientes, en paralelo se colocará un diodo. El control del circuito se llevará a cabo empleando un transistor bjt, tipo npn BD237. De esta forma el Arduino DUE, podrá accionar las válvulas en función de la señales recibidas por los dos sensores de nivel (pin 48 y 52).

En la figura siguiente se muestra el conexionado realizado con el Arduino DUE y el circuito necesario para conseguir el funcionamiento de las válvulas.

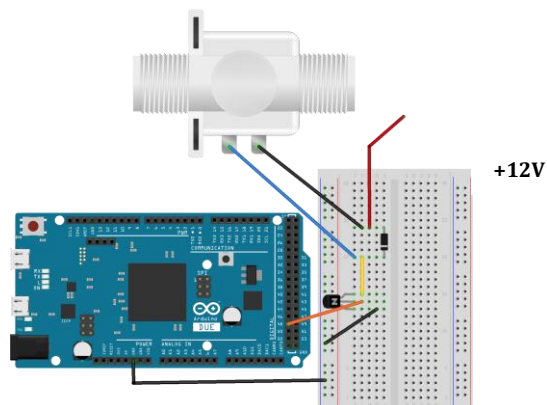


Fig34. Conexionado de una electro-válvula con el Arduino DUE.

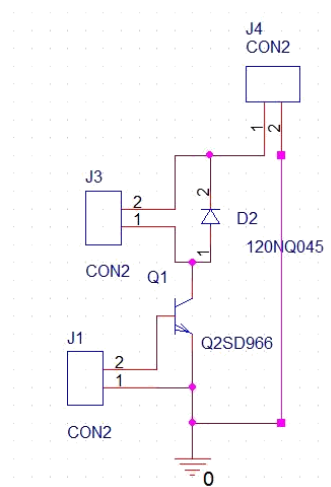


Fig35. Esquema del circuito de una electro-válvula.

1.5.3.7 SENSOR DE NIVEL DE AGUA

Los sensores de nivel del agua serán los encargados de controlar la apertura o cierre de las electro válvulas. Como se puede ver en la figura la entrada de datos de los sensores de nivel se realizará por los pines digitales 32 y 33.

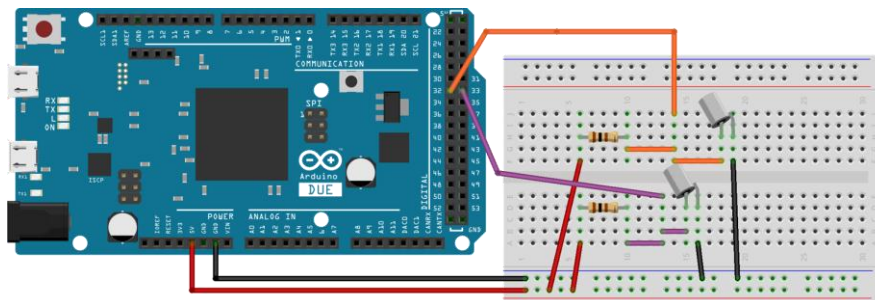


Fig36. Conexión de los sensores de nivel con la placa Arduino DUE.

1.5.3.8 SENSOR DE TURBIDEZ

Este sensor nos ofrece dos opciones de uso, la forma analógica, la cual se ha escogido para la pecera y la forma digital. Si se emplea la opción analógica, la tensión de salida que se obtendrá variará entre 0-4.5V. Además, este valor, se reducirá en cuanto mayor sea la turbidez del agua. Por el contrario, en caso de utilizar la opción digital, la salida será a nivel alto o bajo y existe un potenciómetro que permitirá regular el valor umbral.

A continuación, en la gráfica se ve el conexionado realizado para conectarlo al Arduino Due, el cual consiste en la conexión de la alimentación VCC(5V), GND y la entrada analógica A3

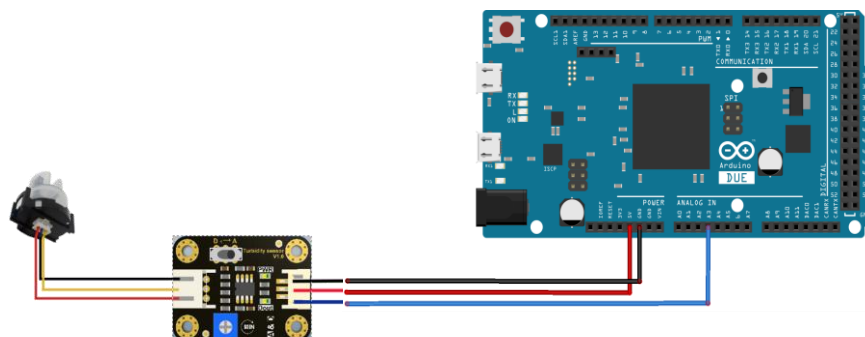


Fig37. Conexión sensor de turbidez con módulo V1.0 a placa Arduino DUE.

Una vez conectado, se han realizado diversas pruebas de medición con diferentes productos de consumo diario. Como se puede ver en la tabla siguiente, el valor de la tensión de salida ha sido mayor en cuanto mayor era la claridad del producto. Una vez obtenidos todos los valores se les ha clasificado en una escala de porcentaje, para mayor facilidad del usuario al parecer en la pantalla LCD.

Se ha partido como valor inicial de la lectura del sensor, con el líquido más claro, es decir con el de mayor valor de tensión, que en este caso ha sido el agua.

	Señal sensor	Volatje(0-5)V	[%]
Leche	15	0,07	98,43
Zumo	309	1,43	67,94
Café	827	4,04	9,42
Agua destilada	917	4,44	0,45
Agua	914	4,46	0,00

Tabla4. Resultados obtenidos con sensor de turbidez.

En la figura siguiente, se puede apreciar a través de una gráfica, el nivel de opacidad obtenida con cada producto : $V_{out} = 100 - \left(\frac{V}{V_o} * 100 \right)$

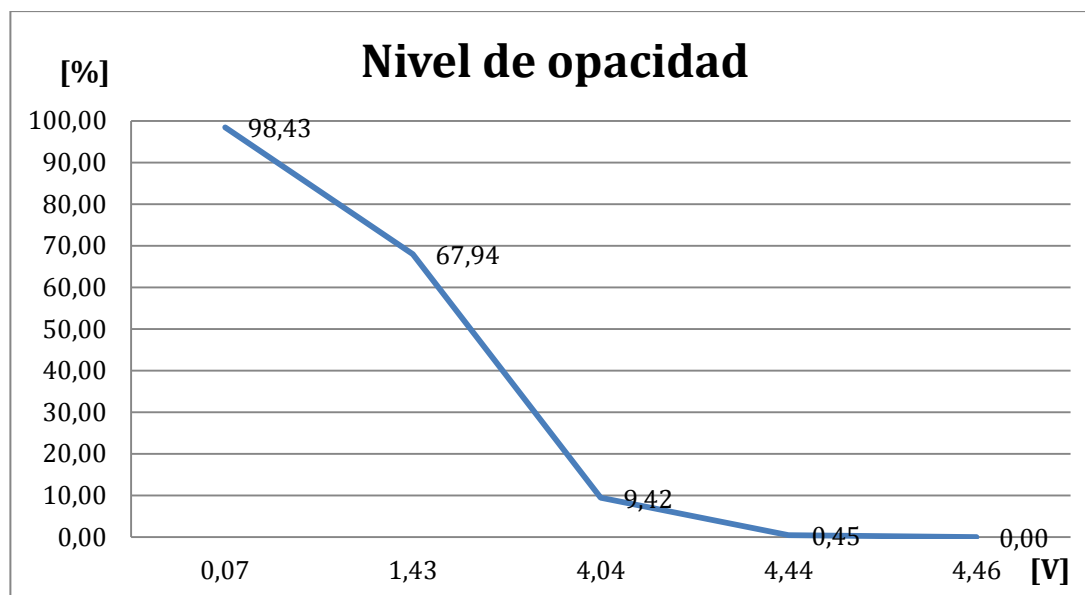


Fig38. Tabla gráfica de los resultados obtenidos con el sensor de turbidez.

Como se puede apreciar en la gráfica, el valor del porcentaje de opacidad de los productos aumenta, según su tensión disminuye. Es decir que la cantidad de solidos suspendidos es mayor y por lo tanto también lo es su nivel de turbidez.

1.5.3.9 MÓDULO BLUETOOTH HC-06

Para la utilización de este módulo, se ha usado un micro controlador Arduino UNO, con el fin de realizar unas pruebas aisladas y verificar su correcto funcionamiento.

Posteriormente para la conexión final, se ha empleado un micro controlador Arduino DUE. Como se puede apreciar en la figura, la placa Arduino UNO solo posee un par de puertos serie(TX-RX), por los que se debe tener cuidado ya que este hecho pueden ocasionar problemas al intentar subir el Sketch.(Problemas detallados en apartados posteriores). Sin embargo, esto no ocurre al usar un Arduino DUE.

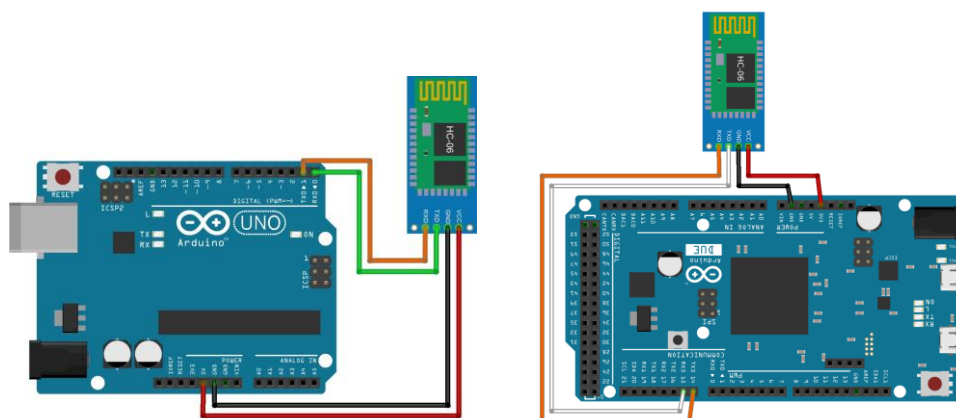


Fig39. Conexionado de módulo HC-06 con Arduino UNO y Arduino DUE.

1.5.3.10 PANTALLA LCD 20X4 CON MÓDULO I2C

La función principal de la pantalla será mostrar los diferentes resultados de los sensores, en función de dos pulsadores. Estos harán la función de botones de “siguiente” o “atrás”. Además junto con los pulsadores, se añadirán 2 resistencias de 10KΩ. Debido a que la pantalla posee un módulo I2C, no tendrá demasiada complicación para conectarse con el micro controlador Arduino DUE, como se puede apreciar en la siguiente figura.

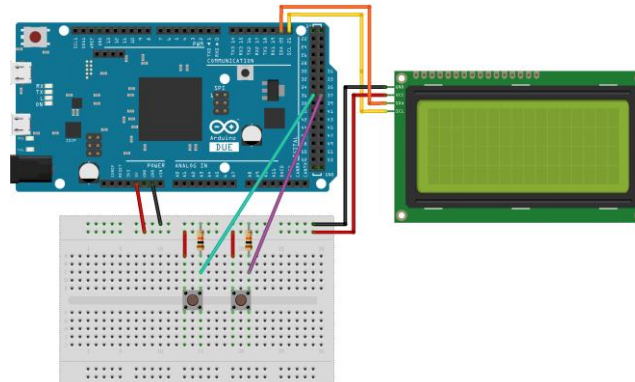


Fig40. Conexión del circuito empleado para el LCD.

1.6 IMPLEMENTACIÓN SOFTWARE

Una vez finalizado el apartado hardware, es decir las partes físicas de la pecera automatizada, se necesita dotar a ésta de la parte software. La parte software abarca toda la parte lógica de este proyecto, y se define como el conjunto de programas e instrucciones que permitirán realizar todas las tareas predefinidas por el usuario. Cabe destacar que en la parte software también se utilizará parte de los cálculos realizados en el apartado “Cálculos hardware”.

En este apartado, se va a definir el módulo bluetooth CH-06 usado para enlazar el microcontrolador Arduino DUE con un móvil smartphone. También se analizará la creación de una aplicación android, mediante el programa MIT APP Inventor, para poder visualizar los datos obtenidos por los sensores LM335 en tiempo real desde el Arduino DUE. Además se detallará el software de control, usado por el microcontrolador y finalmente la programación en lenguaje de alto nivel para llevar a cabo cada tarea.

1.6.1 MODULO BLUETOOTH HC-06

La conexión del módulo bluetooth HC-06 con el microcontrolador Arduino, se puede llevar a cabo de diferentes modos, siempre y cuando se cumpla que:

Módulo Bluetooth	Micro-controlador
Pin VCC	Pin +5V
Pin GND	Pin GND
Pin RXD	Pin TX
Pin TXD	Pin RX

Tabla5. Conexiones módulo HC-06 con un micro controlador Arduino.

Una vez realizado el conexionado del módulo bluetooth con el micro-controlador Arduino, podemos pasar a la programación software. Previamente se han debido tener en cuenta, los problemas que pueden surgir por los puertos serie que este utiliza para comunicarse con el PC. (En el apartado “problemas hallados” se explica estos problemas detalladamente.)

A través de la programación Arduino, pasamos a configurar el nombre y el pin de este módulo. (Código detallado en el apartado “cálculos Software”). Para enlazar el teléfono por primera vez con el módulo bluetooth, deberemos realizar los siguientes pasos:

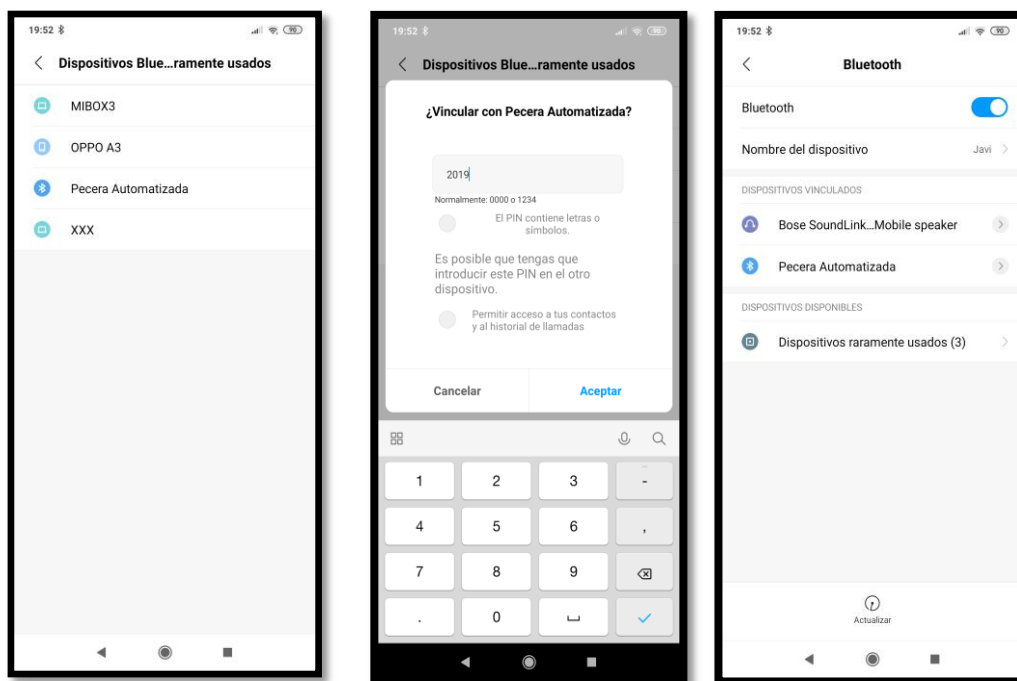


Fig 41. Fases del proceso de vinculación con módulo bluetooth HC-06.

Activar el módulo bluetooth del teléfono y buscar el nombre del dispositivo, en este caso se le ha asignado el nombre de “Pecera automatizada”. (Primera figura) Una vez seleccionado el dispositivo, pedirá la contraseña, que en este caso es “2019”. (Segunda figura) Por último al pasar unos minutos se puede ver que “Pecera automatizada”, forma ya parte de los dispositivos vinculados. (Tercera figura)

1.6.2 APLICACIÓN MOVIL

Tras vincular el dispositivo móvil con el módulo bluetooth HC-06, ya se puede realizar el siguiente paso. Se trata de utilizar una aplicación móvil, compatible con cualquier smartphone android, con el fin de visualizar datos obtenidos por los tres sensores LM335 empleados en la pecera automatizada.

Para ello se ha empleado el programa MIT APP Inventor, el cual es un entorno de desarrollo software creado por Google. Se trata de un software libre y accesible a todo el mundo, con el cual se puede obtener una aplicación “.apk” y además la opción de descarga mediante un código QR.

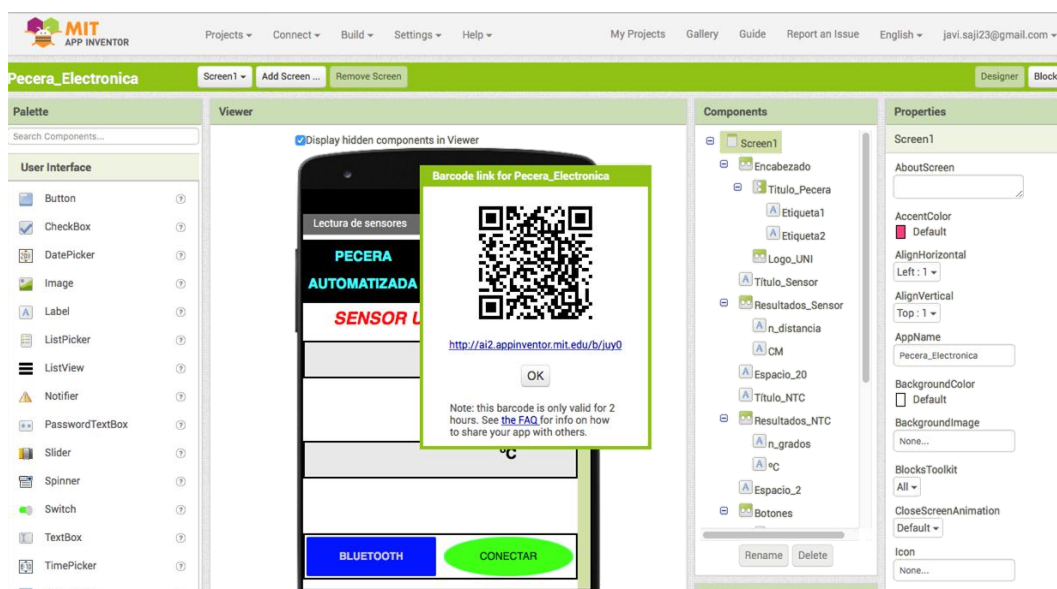


Fig42. Descarga de aplicación final mediante código QR.



Fig43. Código QR disponible solo durante 2 horas después de la solicitud.

La aplicación se divide básicamente en dos herramientas, en la parte superior derecha se puede encontrar la pantalla de “Designer” y la pantalla “Blocks”.

Mediante esta primera pantalla se va a construir la parte visual de la app, es decir se debe elegir una interfaz de usuario a través de la herramienta “Palette”, situada en la esquina superior izquierda. Esta herramienta proporciona todos los medios posibles básicos como cuadros de texto, posibilidad de subir imágenes, audios, enlaces con direcciones webs...etc.

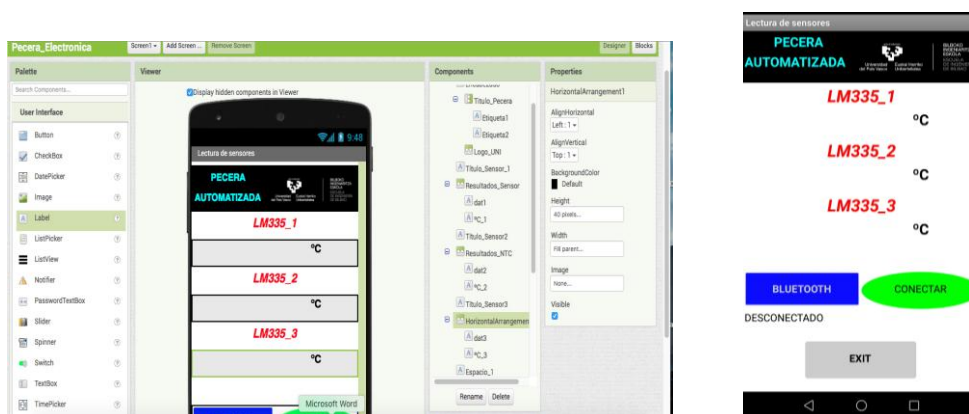


Fig44. Interfaz final de aplicación.

La herramienta designer te da mucha libertad para elegir entre varias opciones, pero tiene un inconveniente y es que solo te da una medida estándar de pantalla para el móvil.

Si esta aplicación se emplea para móviles de pantalla con medidas superiores, seguramente no encaje al completo. Una forma de solucionar este problema es mediante programación en la herramienta de “Blocks”, o calcular una vez descargada un primer borrador de la aplicación, y mediante la introducción de espacios horizontales mover los bloques ya situados.

Una vez obtenida la interfaz, se pasa a la herramienta “Block”, en ella se debe configurar la finalidad que queremos dar a la aplicación. Se trata de de una programación mediante bloques, pero para la cual es necesario tener nociones básicas de programación.

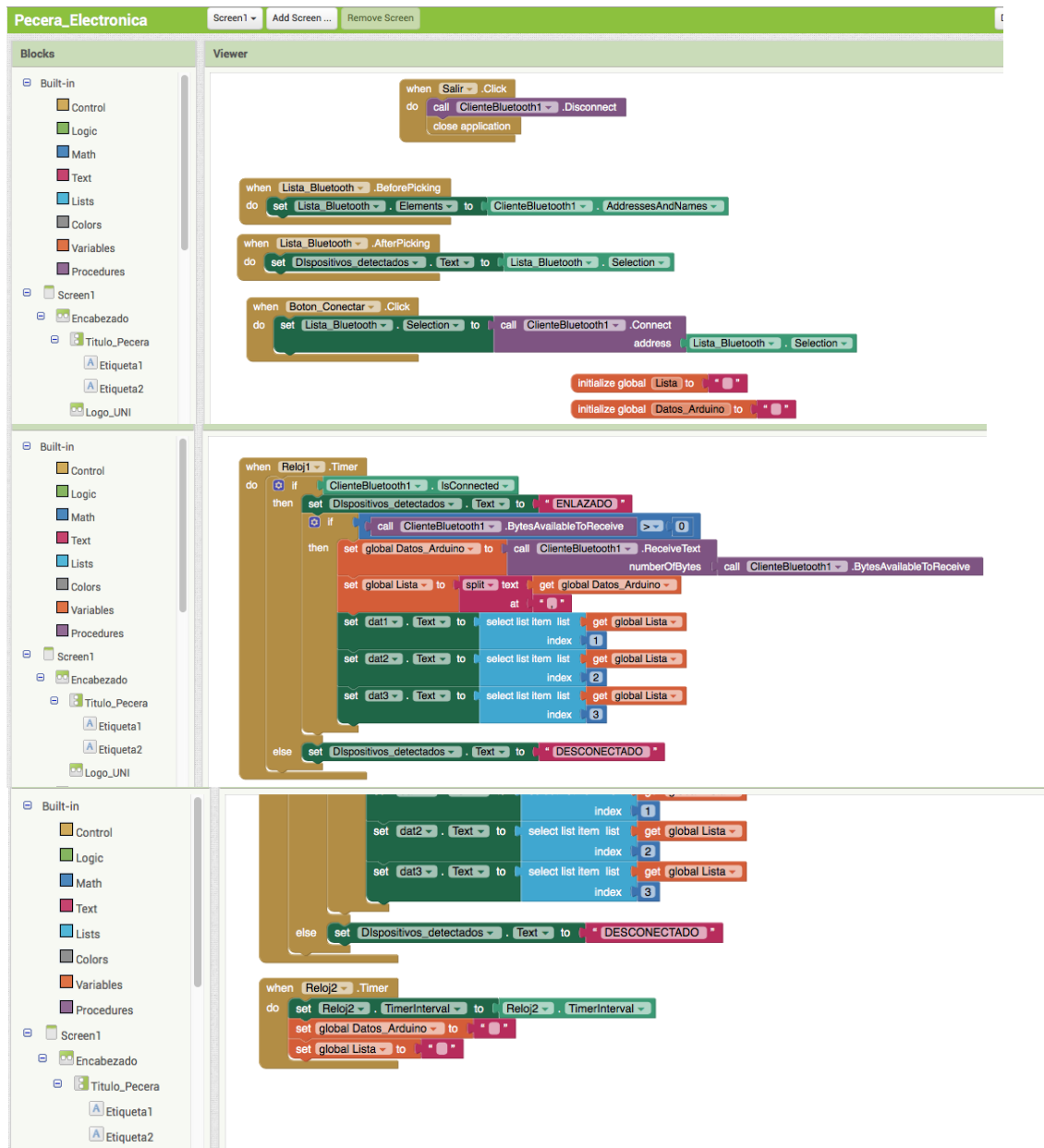


Figura 45. Bloques de programación empleados para la app.

Como se puede observar en las imágenes, al inicio la programación se configura el boton de “salir”, y se asegura que a la vez se desconecta el enlace bluetooth. A continuación, se configura el bluetooth, y después el boton de conectar.

Seguidamente, se introducen las dos variables “Lista” y “Datos Arduino”, las cuales van a jugar un papel importante. Estas variables se van a encargar de adquirir los datos que obtiene el micoporcesador Arduino DUE y las organizará mediante un orden asignado. Para ello se necesita que de Arduino las variables que salgan de las LM335 también tengan un orden determinado.

Finalmente, con la ayuda de dos clock, se van a refrescar estos datos cada cierto tiempo.

1.6.3 PROGRAMACIÓN CON ARDUINO

El entorno de programación Arduino es un IDE (Integrated development environment). Se trata de un programa informático, compuesto por un conjunto de herramientas de programación el cual permite también la conexión USB, para cargar los programas (SKetchs) y comunicarse con ellos.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación. Esto consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware a través del puerto serie.

Entre las funciones más importantes destacan las siguiente opciones:

- Nuevo y Abrir: permite crear y abrir un sketch anteriormente guardado.
- Proyecto: permite abrir un conjunto de sketches determinado que conforman un proyecto concreto.
- Ejemplos: se disponen ejemplos de sketch sobre el uso de la placa de Arduino.

En el menú “editar” se encuentran las opciones básicas de edición, además de copiar el texto como html. En el menú programa, existen las opciones de verificar y compilar, incluir librerías y mostrar la carpeta del programa.

La estructura correspondiente a un código de Arduino atiende a la siguiente forma: en primer lugar el **void setup()** donde se indica el modo de funcionamiento de los pines (entrada y salida), comunicación serie, declaración de variables etc... A

continuación se encuentra el bloque del programa principal en el que se definen las acciones que se espera que realice el programa, **void loop()**. La función `loop()` realiza precisamente lo que sugiere su nombre, se ejecuta de forma cíclica, lo que posibilita que el programa este respondiendo continuamente ante los eventos que se produzcan.

1.6.4 CÁLCULOS SOFTWARE

La realización de los cálculos software, consisten en la programación en alto nivel aceptada por la plataforma Arduino. En su formato real, toda la programación se ha juntado en un solo programa. Sin embargo, a continuación se describen por separado la programación de cada componente con un diagrama de flujo previo al código. Vamos a suponer que las librerías iniciales están ya colocadas desde un principio en todos los casos, por lo que se empezará a describir a partir de la configuración de los pines.

En la siguiente figura, se puede ver el diagrama de flujo general del funcionamiento de la pecera automatizada.

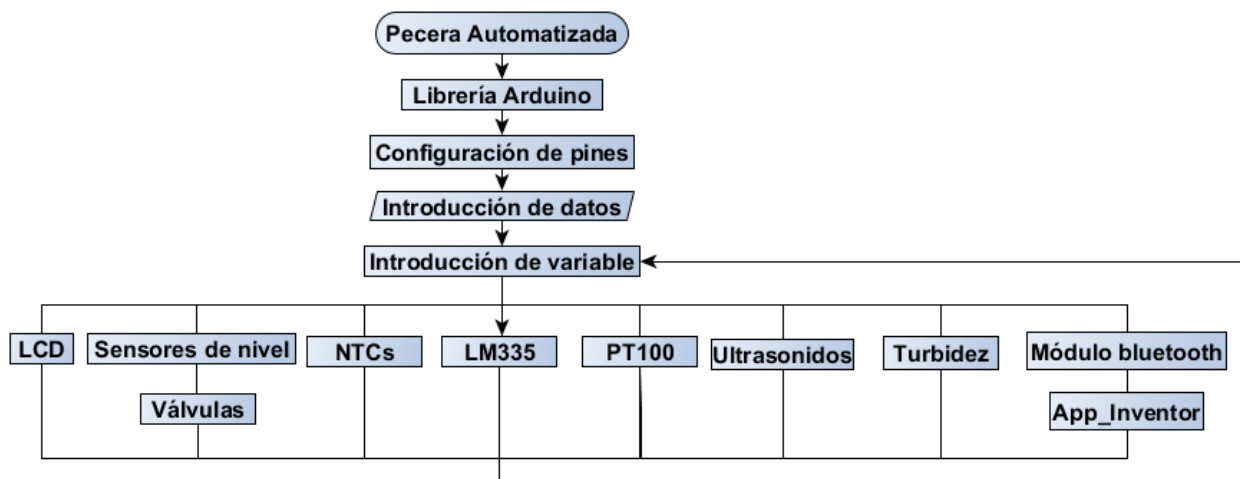


Fig46. Diagrama correspondiente al funcionamiento general de la pecera automatizada.

LIBRERÍA

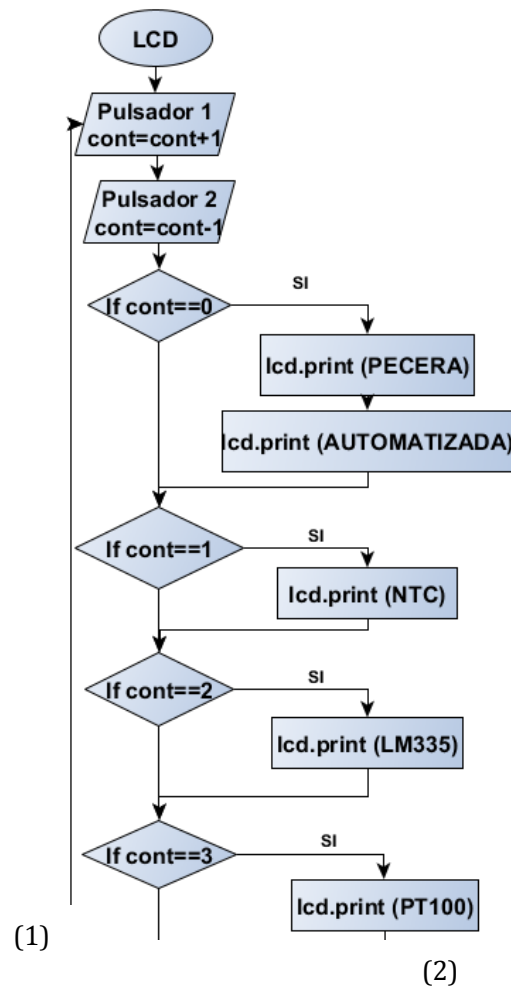
```
#include <math.h>
```

```
// Librería para realizar operaciones aritméticas básicas
```

```
#include <stdio.h> // Standard Input-output header- Configuración para entradas y salidas
```

PANTALLA LCD

A continuación se va a describir el diagrama de flujo de la pantalla LCD, la cual va estar controlada por dos pulsadores, los cuales permitirán elegir la información que se quiera ver. Los dos botones cumplirán la función de avance y retroceso.



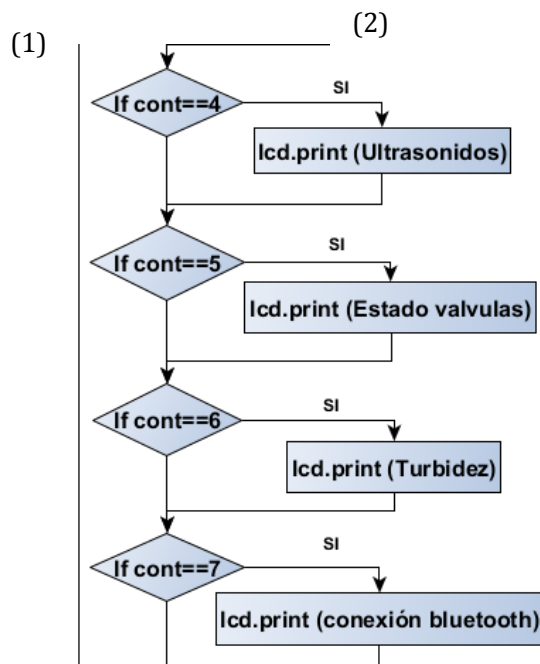


Fig47. Diagrama asociado a la pantalla LCD.

```

#include <LiquidCrystal_I2C.h>           //Librería para controlar LCD 20x4 con módulo I2C
LiquidCrystal_I2C lcd(0x27,20,4);       // Dirección de bus I2C 0x27, 20 columnas, 4 filas.
int pulsador = 36;                      // Asociación de pulsador al pin digital 36
int pulsador2 = 37;                    // Asociación de pulsador al pin digital 37
int cont=0;                             // Asignación de valor 0 a cont que se utilizará en la LCD
  
```

void setup()

```

{
  lcd.init();                          // Iniciación de LCD
  lcd.backlight();                     // Activación de las luces de fondo de LCD
  Serial.begin (9600);                 // Iniciación de la comunicación serial

  pinMode(pulsador, INPUT);            //Asignación de entrada a pulsador
  pinMode(pulsador2, INPUT);           //Asignación de entrada a pulsador2
}
  
```

Void loop

```

{
  int retardo=300;

  int puls=digitalRead(pulsador);           //Lectura del pulsador 1
  if (puls == HIGH)                          //Si el pulsador 1 esta activado
  {
    cont=cont+1;                             //Se suma un valor al actual
    Serial.println("sumas");
  }

  int puls2=digitalRead(pulsador2);          //Lectura del pulsador 2
  if (puls2 == HIGH)                         //Si el pulsador 2 está activado
  {
    cont=cont-1;                             //Se resta un valor al actual
    Serial.println("restas");
  }

  Serial.print("% Contador: ");
  Serial.println(cont);
  delay(1500);
  if(cont==1)                               //Si la variable contador es 1
  {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("NTC1: ");lcd.print(TempNtc1);lcd.print(" C");
    lcd.setCursor(0,1);
    lcd.print("NTC2: ");lcd.print(TempNtc2);lcd.print(" C");
    lcd.setCursor(0,2);
    lcd.print("NTC3: ");lcd.print(TempNtc3);lcd.print(" C");
    delay(retardo);
  }

  if(cont==2)                               //Se la variable contador es 2
  {
    lcd.clear();
    lcd.setCursor(0,0);
  }
}

```

```

    lcd.print("LM335_1: "); lcd.print(TempLm1); lcd.print(" C");
    lcd.setCursor(0,1);
    lcd.print("LM335_2: "); lcd.print(TempLm2); lcd.print(" C");
    lcd.setCursor(0,2);
    lcd.print("LM335_3: "); lcd.print(TempLm3); lcd.print(" C");
    delay(retardo);
  }
  if(cont==3)                                     //Si la variable contador es 3
  {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("PT100_1: "); lcd.print(TempPt1,1); lcd.print(" C");
    lcd.setCursor(0,1);
    lcd.print("PT100_2: "); lcd.print(TempPt2,1); lcd.print(" C");
    lcd.setCursor(0,2);
    lcd.print("PT100_3: "); lcd.print(TempPt3,1); lcd.print(" C");
    delay(retardo);
  }
  if(cont==4)                                     //Si la variable contador es 4
  {
    lcd.clear();
    lcd.print("Dista_US: "); lcd.print(distanceCm); lcd.print("cm");
    delay(1000);
  }
  if(cont==5)                                     //Si la variable contador es 5
  {
    lcd.clear();
    lcd.print("Estados valvulas: ");
    if (valueALTO == 1 && valueBAJO == 1)
    {
      lcd.print("valvula abajo abierta");
    }
  }

```

```
    if (valueALTO == 1 && valueBAJO == 0)
    {
        lcd.print ("Error en Valvulas");
        lcd.print ("\n");
    }
    if (valueALTO == 0 && valueBAJO == 1)
    {
        lcd.print("Proceso de llenado");
    }
    if (valueALTO == 0 && valueBAJO == 0)
    {
        lcd.print("valvula arriba abierta");
    }
    if(cont==6)    //Si la variable contador es 6
    {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Nivel de opacidad del agua");
        lcd.setCursor(0,1);
        lcd.print(porcturb);lcd.print(" % ");
        delay(retardo);
    }
    if(cont==7){cont=6; delay(retardo);}           //Si la variable contador es 7
    if(cont<=0){cont=0; delay(retardo);}
    if(cont==0)                                     //Si la variable contador es 0
    {
        lcd.clear();
        lcd.setCursor(1,0);
        lcd.print("P E C E R A ");
        lcd.setCursor(5,2);
        lcd.print("AUTOMATIZADA");
        delay(retardo);
    }
```

SENSORES DE NIVEL-VÁLVULAS

En la siguiente figura se describe el código de funcionamiento en el micorcontrolador, para conseguir la relación entre los dos sensores de nivel y las electro-válvulas.

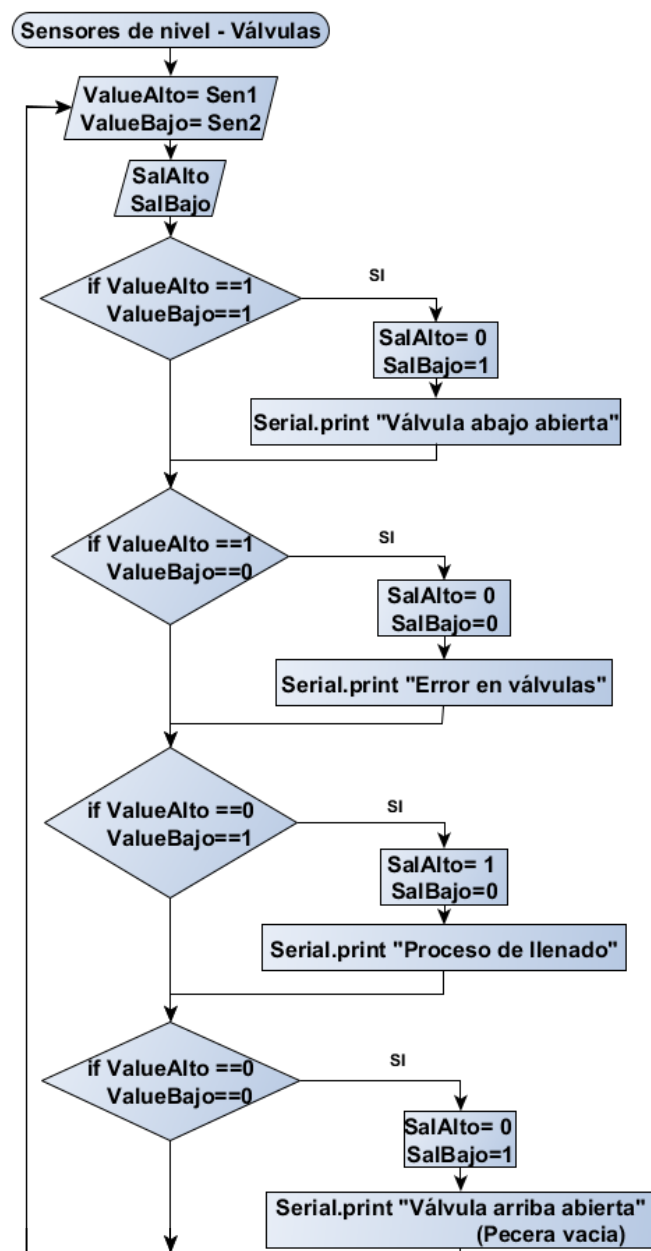


Fig48. Diagrama de los sensores de nivel y válvulas.


```

int Sen1 = 32;           // Asociación de sen1 al pin digital 32
int Sen2 = 30;           // Asociación de sen2 al pin digital 30
int valueALTO = 0;       // Asignación de valor 0 a valueALTO
int valueBAJO = 0;       // Asignación de valor 0 a valueBAJO
int salALTO = 48;        // Asociación de salALTO al pin digital 48
int salBAJO = 52;        // Asociación de salBAJO al pin digital 52

void setup()
{
  pinMode (Sen1, INPUT);    //Asignación de entrada a sen1
  pinMode (Sen2, INPUT);    //Asignación de entrada a sen2
  pinMode (salALTO, OUTPUT); //Asignación de salida a salALTO
  pinMode (salBAJO, OUTPUT); //Asignación de salida a salBAJO
}

void loop()
{
  valueALTO = digitalRead (Sen1); //Lectura del sensor1
  valueBAJO = digitalRead (Sen2); //Lectura del sensor2
  if (valueALTO == 1 && valueBAJO == 1) //Si el sensor de arriba y de abajo estan activados
  {
    digitalWrite(salALTO, LOW);      //Válvula de arriba cerrada
    digitalWrite(salBAJO, HIGH);     //Válvula de abajo abierta
    Serial.println("valvula abajo abierta"); //Imprimir en 'monitor serie'
  }

  if (valueALTO == 1 && valueBAJO == 0) //Si el sensor de arriba está activado y el de abajo
                                     // desactivado
  {
    digitalWrite(salALTO, LOW);      //Válvula de arriba cerrada
    digitalWrite(salBAJO, LOW);      //Válvula de abajo cerrada
    Serial.print ("Error en Valvulas"); //Imprimir 'Error en Válvulas'
    Serial.print ("\n");
  }

  if (valueALTO == 0 && valueBAJO == 1) //Si el sensor de arriba y de abajo están
                                     // desactivados
  {
    digitalWrite(salALTO, HIGH);     //Válvula de arriba abierta
    digitalWrite(salBAJO, LOW);      //Válvula de abajo cerrada
  }
}

```

```

    Serial.println("Proceso de llenado");
  }
  if (valueALTO == 0 && valueBAJO == 0) //Si losl sensor de arriba y de abajo están
                                         //desactivados
  {
    digitalWrite(salALTO, HIGH);          //Válvula de arriba abierta
    digitalWrite(salBAJO, LOW);           //Válvula de abajo cerrada
    Serial.println("valvula arriba abierta");
  }
  delay(1000);
}

```

TERMISTOR-NTC

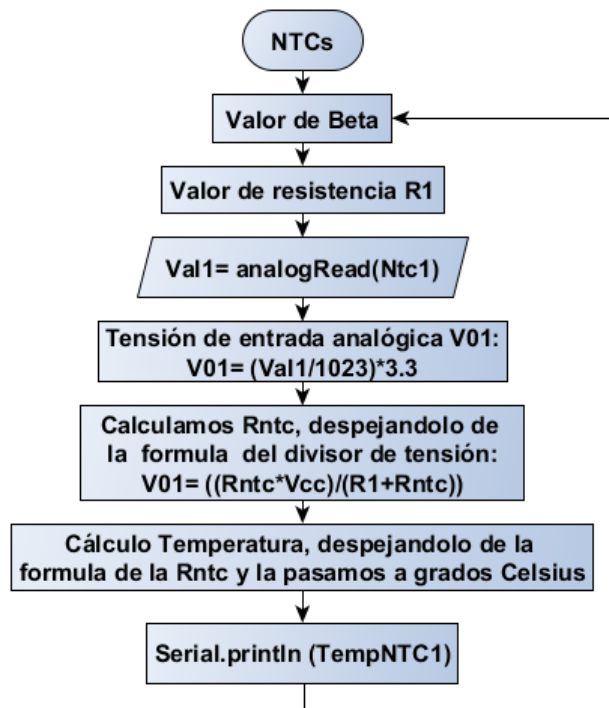


Fig49. Diagrama del funcionamiento de los transductores NTCs.

```

int ntc1=A0;           // Asociación de ntc1 al pin analógico A0
int ntc2=A1;           // Asociación de ntc2 al pin analógico A1
int ntc3=A2;           // Asociación de ntc3 al pin analógico A2

```

```

int B_1=3950;           // Valor de Beta (valor constante) para ntc1
int B_3=3950;           // Valor de Beta (valor constante) para ntc2
int B_2=3380;           // Valor de Beta (valor constante) para ntc3
int R1=9300;            // Valor de R1 en el divisor de tensión para ntc1
int R2=9300;            // Valor de R2 en el divisor de tensión para ntc2
int R3=8500;            // Valor de R3 en el divisor de tensión para ntc3
float Vcc = 5.11;       //Tensión de salida de la fuente
int R0 = 10000;         // Valor de la resistencia de referencia de la ntc a la temperatura de referencia(25°C)
int T0=298;             // Valor de la temperatura de referencia en Kelvins (298-273=25)
  
```

void setup()

```

{
  pinMode(ntc1, INPUT);      //Asignación de entrada a ntc1
  pinMode(ntc2, INPUT);      //Asignación de entrada a ntc2
  pinMode(ntc3, INPUT);      //Asignación de entrada a ntc3
}
  
```

void loop()

```

{
  float val1=analogRead(ntc1);           //Lectura de la NTC1 por el pin analógico
  float vo1=((val1/1023)*3.3)-0.02;       //Cálculo del valor de tensión de la NTC1
  float Rntc1=(R1*vo1)/(Vcc-vo1);        //Despejamos Rntc de la fórmula del divisor de tensión
  float TempNtc1=(B_1/((log(Rntc1/R1)+(B_1/298))))-273 ; //Despejamos Tm de la
                                                    // fórmula de la NTC1 (Rtm)

  float val2=analogRead(ntc2);           //Lectura de la NTC2 por el pin analógico
  float vo2=((val2/1023)*3.3)-0.02;       //Cálculo del valor de tensión de la NTC2
  float Rntc2=(R2*vo2)/(Vcc-vo2);        //Despejamos Rntc de la fórmula del divisor de tensión
  float TempNtc2=(B_2/((log(Rntc2/R2)+(B_2/298))))-273; //Despejamos Tm de la fórmula
                                                    // de la NTC2 (Rtm)

  float val3=analogRead(ntc3);           //Lectura de la NTC3 por el pin analógico
  float vo3=((val3/1023)*3.3)-0.02;       //Cálculo del valor de tensión de la NTC3
  float Rntc3=(R3*vo3)/(Vcc-vo3);        //Despejamos Rntc de la fórmula del divisor de tensión
  float TempNtc3=(B_3/((log(Rntc3/R3)+(B_3/298))))-273 ; //Despejamos Tm de la
                                                    //fórmula de la NTC3 (Rtm)
}
  
```

```

    Serial.print ("\n");
    Serial.print("% NTC1: ");
    Serial.println(TempNtc1);           // Impresión del valor de la temperatura de NTC1
    Serial.print ("\n");
    Serial.print("% NTC2: ");
    Serial.println(TempNtc2);           // Impresión del valor de la temperatura de NTC2
    Serial.print ("\n");
    Serial.print("% NTC3: ");
    Serial.println(TempNtc3);           // Impresión del valor de la temperatura de NTC3
    Serial.print ("\n");
  }

```

LM335

Se presenta la programación perteneciente a los sensores LM335

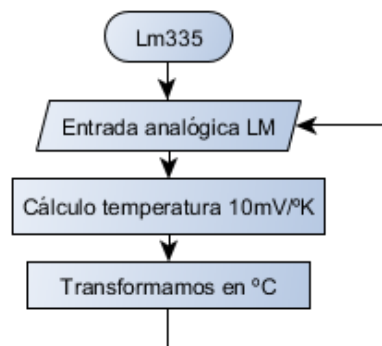


Fig50. Diagrama correspondiente a los sensores LM335.

```

int lm1=A6;           // Asociación de lm1 al pin analógico A6
int lm2=A7;           // Asociación de lm2 al pin analógico A7
int lm3=A8;           // Asociación de lm3 al pin analógico A8
void setup()
{
  pinMode(lm1, INPUT); //Asignación de entrada a lm1
  pinMode(lm2, INPUT); //Asignación de entrada a lm2

```

```

    pinMode(lm3, INPUT);          //Asignación de entrada a lm3
  }

  void loop()
  {
    float TempLm1K = analogRead(lm1) * 0.00322 * 100; // Temperatura Lm1 (Kelvin) con
                                                    //relación 10mV/K - 3.3V

    float TempLm1 = TempLm1K - 268; // Transformación a grados centígrados (-273)

    float TempLm2K = analogRead(lm2) * 0.00322 * 100; // Temperatura Lm2 (Kelvin) con
                                                    //relación 10mV/K - 3.3V

    float TempLm2 = TempLm2K - 269; // Transformación a grados centígrados (-273)

    float TempLm3K = analogRead(lm3) * 0.00322 * 100; // Temperatura Lm3 (Kelvin) con
                                                    //relación 10mV/K - 3.3V

    float TempLm3 = TempLm3K - 270; // Transformación a grados centígrados (-273)
  }

```

RTD-PT100

En el siguiente diagrama se muestra el programa a realizar por las RTD de la pecera automatizada.

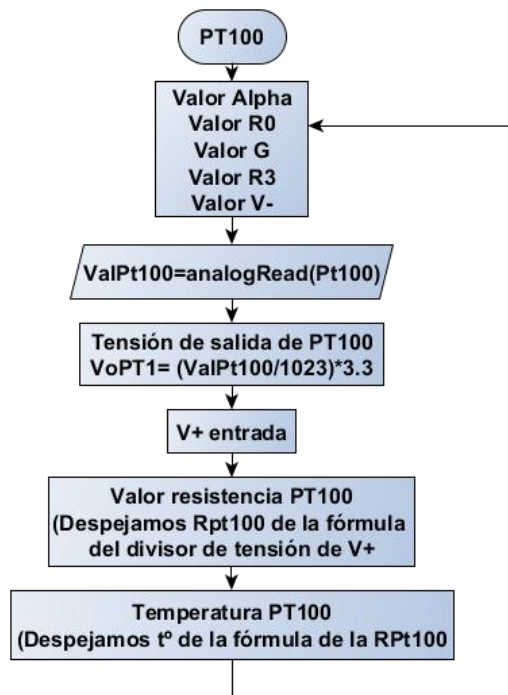


Fig51. Diagrama vinculado a los sensores PT100.

```

int Pt1=A9;           // Asociación de Pt1 al pin analógico A9
int Pt2=A10;          // Asociación de Pt2 al pin analógico A10
int Pt3=A11;          // Asociación de Pt3 al pin analógico A11
float VmenosPt=0.151; // Valor de V- en la PT100 151.15mV
int G=100;             // Valor de Ganancia al pasar por el Amplificador Operacional.
int Ro=100;            // Valor de la resistencia de referencia de la Pt100
float alpha = 0.00385; // Coeficiente de temperatura del conductor a 0°C
int R=3200;            // Valor de la resistencia R3

void setup()
{
  pinMode(Pt1, INPUT); //Asignación de entrada a Pt1
  pinMode(Pt2, INPUT); //Asignación de entrada a Pt2
  pinMode(Pt3, INPUT); //Asignación de entrada a Pt3
}

void loop()
{
  float VoPt1 = (analogRead(Pt1) * 0.00322265625)-0.03; // Lectura del voltaje de PT100_1
  float VmasPt1 = ((VoPt1+(VmenosPt * G))/G); // Cálculo de V+ con la relación de la
                                              // ganancia del A.O.I.
  float RPt1 = (R*VmasPt1)/(Vcc-VmasPt1); // Cálculo de la resistencia de la PT100_1.
  float TempPt1 = (((RPt1/Ro)-1)/alpha)+7.5; // Temperatura (°C) despejados de la
                                              // fórmula de la PT100_1

  float VoPt2 = (analogRead(Pt2) * 0.00322265625)-0.03; // Lectura del voltaje de
                                              // PT100_2
  float VmasPt2 = ((VoPt2+(VmenosPt * G))/G); // Cálculo de V+ con la relación de la
                                              // ganancia del A.O.I.
  float RPt2 = (R*VmasPt2)/(Vcc-VmasPt2); // Cálculo de la resistencia de la PT100_2
  float TempPt2 = (((RPt2/Ro)-1)/alpha)+7.5; // Temperatura (°C) despejados de la
                                              // fórmula de la PT100_2

  float VoPt3 = (analogRead(Pt3) * 0.00322265625)-0.03; // Lectura del voltaje de PT100_3
  float VmasPt3 = ((VoPt3+(VmenosPt * G))/G); // Cálculo de V+ con la relación de la

```

```

//ganancia del A.O.I.

float RPt3 = (R*VmasPt3)/(Vcc-VmasPt3); // Cálculo de la resistencia de la PT100_3
float TempPt3 = (((RPt3/Ro)-1)/alpha)+7.5; // Temperatura (°C) despejados de la
// fórmula de la PT100_1

Serial.print("% PT100_1: ");
Serial.println(TempPt1); // Impresión del valor de la temperatura de la PT100_1
Serial.print("\n");
Serial.print("% PT100_2: ");
Serial.println(TempPt2); // Impresión del valor de la temperatura de la PT100_2
Serial.print("\n");
Serial.print("% PT100_3: ");
Serial.println(TempPt3); // Impresión del valor de la temperatura de la PT100_3
delay(1000);
}

```

SENSOR DE ULTRASONIDOS

En el siguiente diagrama se describe la programación del ultrasonidos, el cual colocado en posición horizontal encima del tope de la pecera podrá medir la distancia a la que se encuentre el agua.

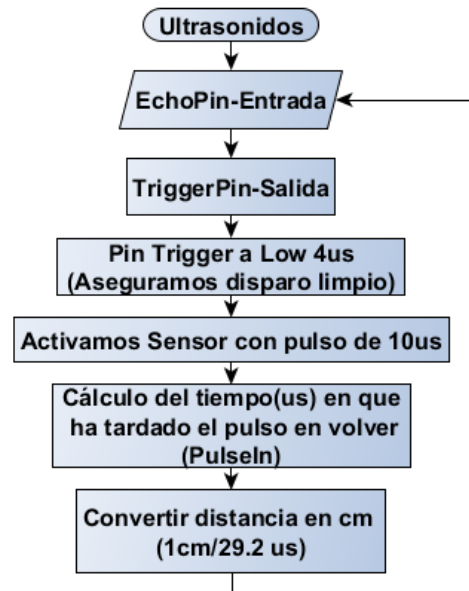


Fig52. Diagrama asociado al sensor de ultrasonidos.

```

const int EchoPin = 6;           // Asociación de EchoPin al pin digital 6
const int TriggerPin = 5;       // Asociación de TriggerPin al pin digital 5
int distancia=0;                 // Asignación de valor 0 a distancia

void setup()
{
  pinMode(TriggerPin, OUTPUT);  //Asignación de salida a turb
  pinMode(EchoPin, INPUT);      //Asignación de entrada a EchoPin
}

Void loop()
{
  long duration, distanceCm;    //Declaración de las variables duration y distaciacm
  digitalWrite(TriggerPin, LOW); //Pin Trigger en estado bajo 4us, para generar un pulso
                                //limpio.

  delayMicroseconds(4);
  digitalWrite(TriggerPin, HIGH); //Pulso eléctrico de 10us
  delayMicroseconds(10);
  digitalWrite(TriggerPin, LOW); //Pin Trigger en estado bajo
  duration = pulseIn(EchoPin, HIGH); //Medición del tiempo que ha tardado el pulso, en
                                    //microsegundos
  distanceCm = (duration * 10 / 292 / 2); //conversión en cm
  Serial.print("Distancia: ");
  Serial.println(distanceCm);
  delay(100);
}

```

TURBIDEZ

En el siguiente apartado se describe el programa relacionado con el sensor de turbidez y el nivel de opacidad a la que se encuentra el agua. Se parte de la variable inicial en el que el agua se considera el líquido con nivel de menor turbidez, siendo el 0%.

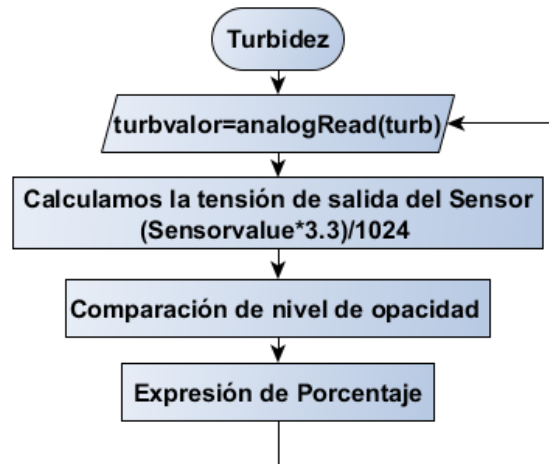


Fig53. Diagrama asociado al sensor de turbidez.

```

int turb=A3;           // Asociación de turb al pin analógico A3
int turbvalor=0;       // Asignación de valor 0 a turbvalor
int porcturb=0;        // Asignación de valor 0 a porcturb

void setup()
{
  pinMode(turb,INPUT);
  Serial.begin(9600); //Baud rate: 9600
}

void loop()
{
  turbvalor=analogRead(turb);
  float voltage = turbvalor * (5 / 1024); // Convert the analog reading
  porcturb= (100-((voltage/4.46)*100)); //Se utiliza 4.46 ya que es el voltaje de nivel del agua
}
  
```

MÓDULO BLUETOOTH

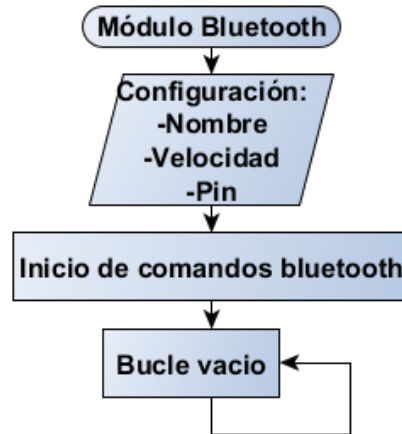


Fig54. Diagrama del módulo bluetooth HC-06.

```

const int LED = 13;           //Asociamos el led 13 que se encuentra dentro del Arduino
const int BTPWR = 12;        //Alimentación del módulo bluetooth por el pin digital 12
char nombreBT[20] = "Pecera Automatizada"; // Asignación de nombre a módulo bluetooth
char velocidad = '4';        //9600 Velocidad con la que se va a trabajar
char pin [5]= "2019";        // Pin elegido para poder realizar el enlace

void setup()
{
  pinMode(LED, OUTPUT);      //Asignación de salida a LED
  pinMode(BTPWR, OUTPUT);    //Asignación de salida a PTPWR
  digitalWrite(LED, LOW);    //Led en estado OFF
  digitalWrite(BTPWR, HIGH); //Alimentación módulo bluetooth en estado ON
  Serial.print("AT");        //Inicio de comando "AT"
  delay(1000);               //Retraso de 1000ms
  Serial.print("AT+NAME");   //Inicio de comando para cambiar nombre
  Serial.print(nombreBT);    //Imprimir nombre
  delay(1000);               //Retraso de 1000ms
  Serial.print("AT+BAUD");   //Inicio de comando para cambiar baudios-velocidad
  Serial.print(velocidad);   //Imprimir valor de velocidad
  delay(1000);               //Retraso de 1000ms
}
  
```

```

Serial.print("AT+PIN");      //Inicio de comando para cambiar pin
Serial.print(pin);           //Imprimir valor de pin
delay(1000);                 //Retraso de 1000ms
digitalWrite(LED, HIGH);     //Led en estado ON
}
void loop()
{

```

APLICACIÓN-MÓVIL

A continuación se describe la programación posterior al enlace con el módulo bluetooth, que necesita la aplicación para poder mostrar los resultados obtenidos con los sensores LM335.



Fig55. Diagrama vinculado a la aplicación utilizada por el dispositivo móvil.

```

float dat1 =0;                // Asignación de valor 0 a dat1, para la conexión bluetooth
float dat2 =0;                // Asignación de valor 0 a dat2, para la conexión bluetooth
float dat3 =0;                // Asignación de valor 0 a dat3, para la conexión bluetooth
char buffer[10];              //Variable que permite almacenar datos de forma ordenada.

```

```

float lectura_lm335_1() //Función para obtener 1er dato de salida de la temperatura de las NTCs
{
    float TempLm1K = analogRead(lm1) * 0.00322 * 100; // Temperatura Lm1 (Kelvin) con
                                                    // relación 10mV/K - 3.3V

    float TempLm1 = TempLm1K - 268; // Transformación a grados centígrados (-273)
    return TempLm1; // Dato que se obtendrá de la función
}

float lectura_lm335_2() //Función para obtener 2ndo dato de salida de la temperatura de las NTCs
{
    float TempLm2K = analogRead(lm2) * 0.00322 * 100; // Temperatura Lm2 (Kelvin)
    con
                                                    //relación 10mV/K - 3.3V

    float TempLm2 = TempLm2K - 269; // Transformación a grados centígrados (-273)
    return TempLm2; // Dato que se obtendrá de la función
}

float lectura_lm335_3() //Función para obtener 3º dato de salida de la tº de las NTCs
{
    float TempLm3K = analogRead(lm3) * 0.00322 * 100; // Temperatura Lm3 (Kelvin)
                                                    //con relación 10mV/K - 3.3V

    float TempLm3 = TempLm3K - 270; // Transformación a grados centígrados (-273)
    return TempLm3; // Dato que se obtendrá de la función
}

Void loop()
{
    float dat1 = lectura_lm335_1(); //Dato de la función 1era
    float dat2 = lectura_lm335_2(); //Dato de la función 2nda
    float dat3 = lectura_lm335_3(); //Dato de la función 3ra
    sprintf(buffer, "%d,%d,%d",dat1,dat2,dat3); //variable que almacena variables de tipo int,
                                                    // char en una sola cadena.

    Serial.println(buffer); //Imprimir dato buffer
    delay(1000);
}
  
```

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA.

TRABAJO FIN DE GRADO

***< PECERA AUTOMATIZADA, MONITORIZADA
POR PANTALLA LCD Y POR MOVIL VIA
BLUETOOTH, BASADA EN LA PLATAFORMA
ARDUINO >***

DOCUMENTO 2- < RESULTADOS OBTENIDOS >

Alumno/Alumna: <Salas, Jiménez, Héctor Javier>

Director/Directora (1): <Oleagordia, Aguirre, Iñigo Javier>

Curso: <2019-2020>

Fecha: < 28, Octubre, 2019>

2. CONCLUSIONES

Hoy en día, la automatización de máquinas y objetos es un tema que cada vez va más en aumento. La mayoría de las veces, consisten en mejoras que pueden facilitar alguna tarea o aportar más datos. En el caso de la pecera automatizada, se quiere realizar un control más detallado del medio que la rodea y añadirle funciones extras para facilitar el llenado. Además se agrega información adicional a la que se podrá tener acceso mediante una pantalla LCD o mediante un dispositivo móvil.

La realización del proyecto, se ha podido llevar a cabo gracias a la combinación de conocimientos en su mayoría teóricos, que se han ido obteniendo a lo largo de la carrera y que ahora se han tenido que poner en práctica. A pesar de que a la pecera no se le ha dado un uso real o comercial, se ha pensado en este proyecto más como un prototipo, para futuros desarrollos. Aun así se ha podido comprobar la eficacia de cada uno de sus componentes hardware con su respectivo software, de manera que puedan cumplir con sus requisitos.

Entre los objetivos cumplidos destacan el conocimiento de dispositivos comerciales en el mercado actual. Para su montaje se ha tenido que realizar un estudio de mercado, comparando los dispositivos entre varias opciones. También se ha realizado un estudio más profundo sobre el microcontrolador Arduino DUE, así como de las muchas posibilidades que nos ofrece.

El montaje de todos los dispositivos ha dado como resultado la correcta unión entre hardware y software, tal y como se propuso desde un principio. Cada componente cumple con su cometido. Además la relación de los diferentes tipos de sensores nos ha permitido realizar una comparación en un mismo entorno y de una misma temperatura. También la visualización a través del móvil de los resultados de los sensores LM335 ha sido un éxito. Cabe destacar que antes es necesario descargarse la aplicación al dispositivo móvil, de lo contrario no se podría visualizar nada.

Como resultado final, se puede decir que el proyecto ha cumplido con todos los objetivos iniciales, de forma correcta. De hecho por el camino algunos objetivos han evolucionado o se han implementado nuevas ideas. También han surgido

problemas a los que se les ha encontrado diversas soluciones, aplicando las más convenientes en cada momento como se describen en apartados posteriores.

2.1 DIAGRAMA DE GANTT

En la tabla siguiente se describen las fases del proyecto y el orden en el que se han ido realizando las distintas tareas. Además con cada tarea se adjunta la fecha de inicio y fin, así como la duración de días referentes a ese periodo.

	PECERA AUTOMATIZADA	INICIO	DURACIÓN	FINALIZACIÓN
1	Documentación previa del proyecto	7-ene	9	16-ene
2	Ideas para definir el proyecto	13-ene	16	29-ene
3	Análisis de posibles componentes hardware	1-feb	8	9-feb
4	Elección del sistema de procesamiento	10-feb	2	12-feb
5	Relación componentes hardware y software	15-feb	5	20-feb
6	Boceto de la unión de todos los componentes	25-feb	7	4-mar
7	Diseño de circuitos de acondicionamiento	11-mar	17	28-mar
8	Cálculos hardware	1-abr	39	10-may
9	Cálculos software	22-abr	44	5-jun
10	Estudio económico del material	27-may	9	5-jun
11	Construcción de circuitos por separados	6-jun	39	15-jul
12	Pruebas iniciales	13-jun	37	20-jul
13	Montaje final del proyecto	22-jul	43	3-sep
14	Agrupación de documentación del proyecto	7-ene	246	10-sep
15	Selección de información más necesaria	10-jul	55	3-sep
16	Redacción del informe final	1-sep	57	28-oct

Tabla 6. Diferentes fases del proyecto.

Con los datos de esta tabla se ha realizado un diagrama de Gantt, en el que se pueden expresar mejor estos resultados. Como se pueden ver, muchas tareas pueden coincidir en periodos de fechas, ya que algunas de ellas se han hecho a la vez. Cabe destacar que la tarea de agrupación de documentación se ha ido realizando casi desde el principio y durante todo el proyecto.

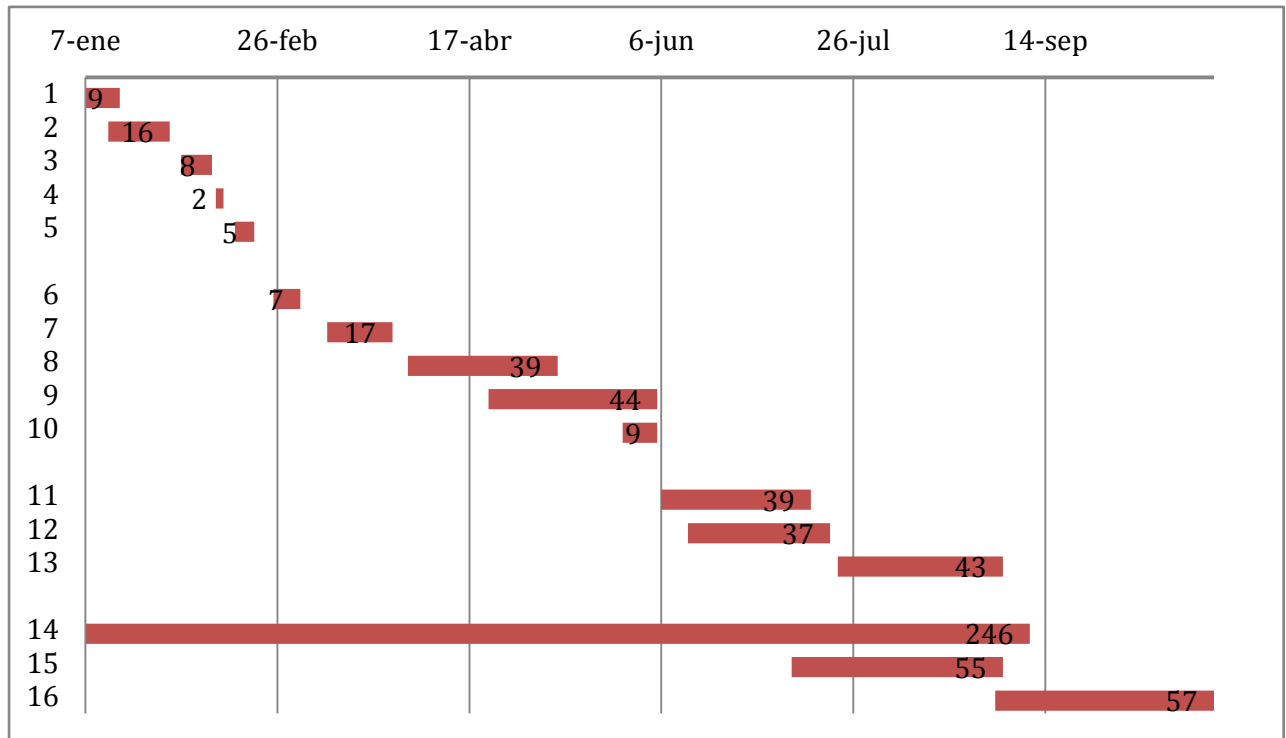


Fig56. Diagrama de Gantt.

2.2 PROBLEMAS HALLADOS

Modulo Bluetooth

Uno de los problemas hallados a lo largo de la realización del proyecto, ha tenido que ver con el módulo bluetooth. Se debe tener en cuenta que dependiendo del tipo de micro controlador usado para la conexión con el módulo bluetooth, puede haber problemas principalmente en los pines de transmisión (TXD) y recepción (RXD).



Fig57.Módulo bluetooth HC-06.

El problema que ha surgido, viene dado debido a que los puertos series (pin 0 y 1), que emplea la placa Arduino para comunicarse con el ordenador, coinciden con los pines de transmisión y recepción de datos del módulo bluetooth HC-06. Como consecuencia de ello en el micro-controlador Arduino UNO, que solo dispone de un único puerto serie de transmisión y recepción de datos, una vez conectado el módulo HC-06, no se podrá recibir o aceptar algún código (sketch) nuevo que se quiera cargar en él. Las soluciones disponibles que se han encontrado ante esta situación han sido:

- Desconectar el módulo HC-06 cada vez que se quiera subir un sketch nuevo o añadir modificaciones.
- Utilizar la librería "SoftwareSerial", la cual nos permite asignar los puertos series a otros pines.
- Emplear un microcontrolador con más puertos series como es Arduino DUE
- Cambiar el pin de alimentación Vcc a un puerto de salida digital, el cual se programa para que el módulo bluetooth HC-06 se desconecte cuando el micro controlador quiera comunicarse con el ordenador.

Finalmente se ha optado por la última de las opciones, ya que de esa forma se asegura un funcionamiento de código en las dos placas del micro controlador.

Electro-válvulas

Otro de los problemas encontrados a lo largo del proyecto, ha sido más de tipo técnico y está relacionado con las electro-válvulas. El problema reside en que las válvulas solenoides tanto si están abiertas o cerradas permiten el paso del agua siempre y cuando se ejerza un mínimo de presión.

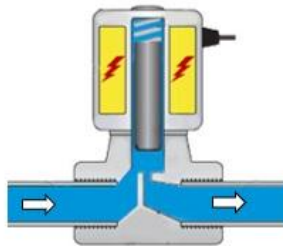


Fig58. Esquema interno de una electroválvula.

Dado que la entrada del agua a la pecera se hace a través de un grifo de agua, este problema no le afecta. Sin embargo, la salida del agua no tenía ninguna presión. La

solución encontrada a este problema fue conseguir una bomba de succión de agua manual para peceras y conectarla a la manguera de salida de esta.



Figura59. Bomba de succión de agua.

Cables conexión

Para el montaje del proyecto, se han utilizado cables machos y hembras habituales

2.3 RESULTADOS

En este apartado se va a definir todos los resultados obtenidos una vez se ha completado el montaje final de la pecera automatizada. Para una mejor descripción se ha dividido la obtención de estos datos en cuatro partes. Por un lado los resultados de temperatura obtenidos por los distintos sensores. A continuación el sensor de turbidez Después el sensor de ultrasonidos, la relación sensores de nivel -apertura y cierre de las electro-válvulas, y por último la recogida de datos a través de un dispositivo móvil. La visualización de todos estos datos se ha hecho mediante una pantalla LCD, por lo que con cada apartado se adjunta una foto de los resultados vistos en ella.

En la siguiente figura se puede apreciar el esquema hardware de la pecera automatizada. En ella se puede ver la posición final de los distintos componentes en relación a la pecera y al Arduino DUE.

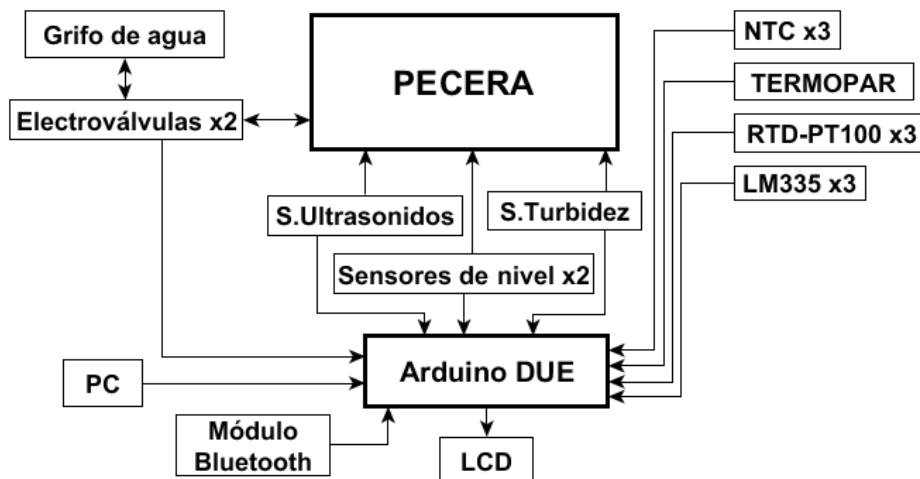


Fig60. Diagrama hardware.

SENSORES DE TEMPERATURA

Los sensores de temperatura se han situado en la parte derecha de la pecera. Exceptuando el termopar tipo K que se utilizará a través del multímetro y que solamente hay un sensor de ese tipo, en todos los demás casos hay tres sensores por cada tipo, por lo que finalmente tendremos diez datos de temperatura. El objetivo principal consiste en recoger la temperatura de un mismo entorno y comparar la precisión de cada sensor. Cabe destacar que se han intentado posicionar lo suficientemente cerca entre ellos, pero evitando que se rocen entre ellos.

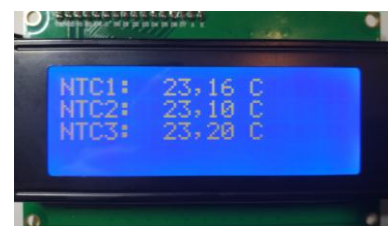
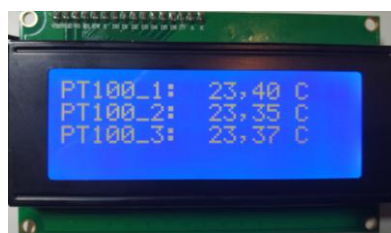
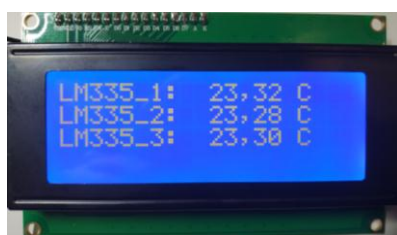


Fig61. Resultados obtenidos de los diferentes sensores de temperatura.

En la figura anterior podemos ver los resultados obtenidos por cada sensor. El único dato que falta es el correspondiente al termopar tipo k cuya temperatura

marcaba los 23.4°C. Cabe destacar que la mayoría de los datos coinciden con la temperatura, con la única diferencia de las décimas. Por lo tanto se pudo apreciar que el acondicionamiento con cada uno de ellos ha sido acertado.

SENSOR DE TURBIDEZ

El sensor de turbidez en la mayoría de los casos, dará el mismo valor de porcentaje de opacidad. El líquido que entra inicialmente en la pecera siempre será agua del grifo, lo que nos asegura una cantidad de partículas suspendidas mínimas y aptas para el consumo humano. Su función está pensada más a largo plazo, en el caso que el agua de la pecera debido a los seres vivos que habiten en ella o por factores externos se haya contaminado. Es entonces cuando el sensor será capaz de detectar y hacer visible este cambio de porcentaje.



Fig62. Porcentaje de opacidad del nivel del agua de la pecera automatizada.

SENSORES DE NIVEL, ULTRASONIDOS Y ELECTRO-VÁLVULAS

Para el correcto funcionamiento de las electro-válvulas, ha sido necesario el uso de dos mangueras metálicas flexibles de ducha. Estas van a conectar por un lado la electro-válvula de entrada con el grifo de agua del fregadero, y la electro-válvula de salida con la bomba de succión de agua que terminará también en el fregadero. Cabe destacar que a pesar de estar automatizada la entrada y salida de agua de la pecera, el paso de agua del grifo no lo está. Por ello, para su uso es necesario controlar la salida manualmente. Esto es debido a que si se mantiene abierto el grifo, se permite el paso del agua. Si la válvula en ese momento está cerrada, la presión del agua con la que sale hace que la manguera flexible no soporte durante

mucho tiempo la presión y acabe saliendo agua por los costados del extremo de la conexión de la manguera.

Otro punto a tener en cuenta es la importancia de la colocación de los sensores de nivel. De la forma en la que se encuentren estos sensores, dependerá que las válvulas estén abiertas o cerradas.

Por último, el sensor de ultrasonidos en este caso actuará como un complemento de refuerzo en este proceso, en el que se podrá calcular la distancia del agua en cada momento. El sensor, estará situado en el tope de la pecera y desde allí medirá la distancia según suba el nivel del agua.

Una vez se han tenido en cuenta estos puntos se procede a la lectura del estado de las válvulas dependiendo de la situación.

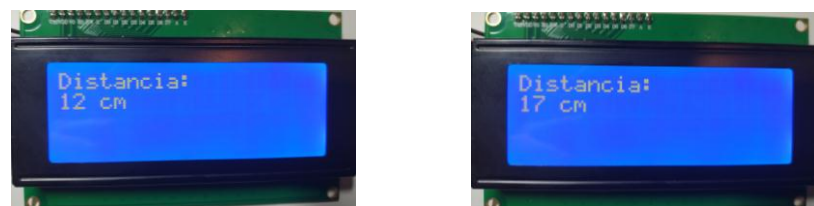


Fig63. Distancia obtenida por el sensor de ultrasonidos según incrementaba el nivel del agua.



Fig64. Estados de las electro-válvulas que se pueden dar según los sensores de nivel.

Se puede ver que se han considerado cuatro estados posibles, aunque en la gran mayoría de los casos solamente se darán tres. El estado que no es muy probable es el de “Estado de error”, ya que para dar lugar en este caso deberá suceder alguna avería en la posición de los sensores de nivel, ya que de lo contrario el aumento del nivel del agua siempre empezará de abajo hacia arriba.

CONEXIÓN CON SMARTPHONE

Como alternativa a la pantalla LCD, se ha prepuesto la lectura de los datos a través de un móvil con la ayuda de una aplicación.

Dado que solo se quiere comprobar que es posible esta conexión con la ayuda del módulo bluetooth HC-06 de Arduino, se ha elegido un tipo de sensor de temperatura para llevar a cabo esta lectura. Concretamente se tratan de los tres sensores LM335 de la pecera automatizada. En caso de realizar una posible mejora con un desarrollo mayor en este proyecto, se podría realizar la configuración para la lectura de todos los demás componentes.

La acción previa a la lectura de los datos de los sensores consiste en descargar la aplicación a través de un código QR. Una vez instalada en el dispositivo seleccionado para esta tarea es necesario realizar el enlace activando el bluetooth del teléfono. Y finalmente el usuario, a través de la aplicación podrá visualizar los resultados en su dispositivo móvil.

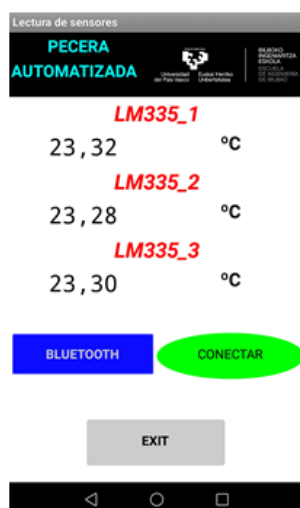


Fig65. Datos obtenidos con la aplicación móvil.

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA.

TRABAJO FIN DE GRADO

***< PECERA AUTOMATIZADA, MONITORIZADA
POR PANTALLA LCD Y POR MOVIL VIA
BLUETOOTH, BASADA EN LA PLATAFORMA
ARDUINO >***

DOCUMENTO 3- <ASPECTOS ECONÓMICOS >

Alumno/Alumna: <Salas, Jiménez, Héctor Javier>

Director/Directora (1): <Oleagordia, Aguirre, Iñigo Javier>

Curso: <2019-2020>

a

Fecha: < 28, Octubre, 2019>

3. ASPECTOS ECONÓMICOS

A continuación, se va a detallar todas las partes del proceso necesarios o que influyeran en el aspecto económico del proyecto.

En este caso no se han tenido en cuenta el material necesario que se haya tenido que reponer en caso de avería, ni los recursos humanos implicados, así como las horas que se hayan empleado en el proceso.

3.1 LISTA DE MATERIALES NECESARIOS

Una vez seleccionada la idea sobre el proyecto a diseñar, fue necesario realizar la elección de los distintos materiales para su montaje. En esta primera lista no se tuvo en cuenta si parte del material tendría que ser comprado o si ya se disponía de él, tanto por cuenta propia o por cuenta de la Universidad.

Fuente de alimentación	Arduino Uno
Osciloscopio	Arduino Due
Protobards	Pantalla LCD
Multímetro	Estaño
Recipiente grande de plástico	Placas soldadura
Electroválvulas	Bornas
Alicates	Cables Hembra-Hembra
Cinta Aislante	Cables Macho-Hembra
Destornilladores	Sensor Turbidez
Silicona	INA 126
Sondas PT100	Soldador
Resistencias NTC	Termómetro
LM 335	Ordenador
Sensor ultrasonido	Diodos
Sensores Nivel	Tubos de salida y entrada de agua
Modulo bluetooth	Resistencias
Transistores	

Tabla7. Componentes iniciales.

Para facilitar la realización del presupuesto, se eliminó de la lista, el material del que ya se disponía o que no ha sido necesario comprarlo y también se subdividió en dos apartados:

- Material electrónico
- Material común

Por un lado, dentro del material electrónico constan todos aquellos dispositivos que intervienen en el funcionamiento de los diferentes circuitos, tanto para recoger información, recibirla o procesarla.

	€/Ud	Nº	Total €
Electroválvulas	9	2	18
Sondas PT100	10	1	10
Resistencias NTC	0,8	3	2,4
LM 335	1	3	3
Sensor ultrasonido	1,22	1	1,22
Sensores Nivel	2,5	2	5
Modulo bluetooth	8,49	1	8,49
Arduino Due	33,95	1	33,95
Pantalla LCD 20x4	15	1	15
Sensor Turbidez	24	1	24
INA 126	2,26	3	6,78
Modulo bluetooth	8,5	1	8,5
Placa Arduino DUE	30	1	30
Amplificador Operacional	0.50	3	1.50

Tabla8. Componentes electrónicos.

Por otro lado tenemos el material común, más relacionado con la construcción de estos circuitos y del proyecto físico en sí.

	€/Ud	Nº	Total €
Rollo de estaño	8	1	8
Placas soldadura	4,5	6	27
Bornas	0,5	20	10
Cables (pack)	2	3	6
Cables Macho-Hembra	2	3	6
Recipiente grande de plástico	6	1	6
Alicates	4	1	4
Cinta Aislante	1,5	2	3
Destornilladores	3	2	6
Silicona	4	1	4
Termómetro	10	1	10

Tabla 9. Material no electrónico

3.1.2 LUGARES DE COMPRA

Principalmente a la hora de comprar material, se optó por tiendas físicas, debido a la cercanía y a la facilidad de compra al instante.

Gran parte del material electrónico se realizó en dos tiendas dedicadas a vender toda clase de dispositivos electrónicos. Y solamente los materiales no tan habituales (sensor de turbidez, electroválvulas...etc.) se compraron en varias casas a través de internet.

3.1.3 COSTES NO INCLUIDOS

Dentro de esta lista final, no se han contemplado los costes indirectos relacionados con el desarrollo y montaje de este proyecto.

Se tratan de gastos relacionados con los lugares de trabajo, material complementario para hacer que nuestro proyecto funcione (fuentes de alimentación, osciloscopio, soldadores...etc) y al que se ha tenido acceso gracias a la universidad. También material que no ha hecho falta comprar porque desde un principio ya se disponía de él.

3.1.4 DISPOSITIVOS DE COMPROBACIÓN

Cabe destacar entre gastos extras, dos dispositivos que no eran tremendamente necesarios, pero para asegurar el correcto funcionamiento de este proyecto se han realizado.

Se tratan de un termómetro digital y de una placa complementaria para el sensor de turbidez. El termómetro se usó para comprobar el nivel de precisión con los distintos sensores encargados de recoger la temperatura ambiente, mientras que la placa del sensor de turbidez fue necesaria para comparar los niveles obtenidos por el circuito de acondicionamiento creado para poder medir los niveles de turbidez.

3.1.5 PRESUPUESTO FINAL

Finalmente haciendo un recuento de todo el material comprado, según las tablas detalladas anteriormente y teniendo en cuenta los gastos no incluidos, se ha llegado a la conclusión de que el coste necesario para haber realizado un proyecto de esta magnitud es de 266,84€.

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA.

TRABAJO FIN DE GRADO

***< PECERA AUTOMATIZADA, MONITORIZADA
POR PANTALLA LCD Y POR MOVIL VIA
BLUETOOTH, BASADA EN LA PLATAFORMA
ARDUINO >***

DOCUMENTO 5- <BIBLIOGRAFÍA>

Alumno/Alumna: <Salas, Jiménez, Héctor Javier>

Director/Directora (1): <Oleagordia, Aguirre, Iñigo Javier>

Curso: <2019-2020>

Fecha: < 28, Octubre, 2019>

4. BIBLIOGRAFÍA

- LIBROS

- [1] Miguel Colom Barco – Apuntes de Instrumentación Electrónica
- [2] Ramón Pallás Areny(1993)-Adquisición y distribución de señales. Barcelona: Marcombo Boixareu Editores.
- [3] Luis Prat Biñas(1998)- Circuitos y dispositivos electrónicos. Barcelona: Ediciones UPC
- [4] Ramón Pallás Areny(1995)- Sensores y acondicionadores de señal. Barcelona: Marcombo Boixareu Editores
- [5]Miguel A. Pérez García, Juan C. Álvarez Antón, Juan C. Campo Rodríguez, Fco. Javier Ferrero Martín, Gustavo J. Grillo Ortega – Instrumentación Electrónica. Thompson
- [6]Miguel Ángel Pérez García(2014). Instrumentación Electrónica. Asturias: Paraninfo.
- [7]José L. Martín González, Jagoba Arias Perez, Unai Bidarte Peraita, Pedro Ibañez Ereño, Jesús Lázaro Arroategui, José L. Martín Gonzalez, Aitzol Zuloaga Izaguirre.(2007).Electrónica Digital. Madrid: Delta Publicaciones.

- DIRECCIONES DE INTERNET

- [8] <https://www.luisllamas.es/medir-temperatura-con-arduino-y-sensor-lm35>
- [9] <https://es.wikipedia.org/wiki/RTD>
- [10] <https://www.zonamaker.com/arduino/modulos-sensores-y-shields/ultrasonido-hc-sr04>
- [11] <https://www.rinconingenieril.es/como-usar-un-termistor-ntc/>
- [12] <http://comofunciona.co.com/un-termistor/>
- [13] <https://es.wikipedia.org/wiki/Termistor>
- [14] <https://www.rinconingenieril.es/medir-temperatura-con-sensor-lm35/>
- [15] <https://prezi.com/mocyoe6dsxl/semiconductores-lm35-lm335/>
- [16] <https://www.jmi.com.mx/pt100>
- [17] <https://srcsl.com/que-es-un-sensor-pt100/>
- [18] <https://es.wikipedia.org/wiki/RTD>

- [19] <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/alimentar-el-arduino-la-guia-definitiva/>
- [20] https://wiki.dfrobot.com/Turbidity_sensor_SKU_SEN0189
- [21] <https://programarfacil.com/blog/arduino-blog/sensor-de-nivel-de-agua-con-arduino/>
- [22] <http://sensorultrasonico.blogspot.com/>
- [23] <https://geekytheory.com/sensor-de-ultrasonidos-hc-sr04-piezoelectrico-arduino>
- [24] <https://www.es.endress.com/es/instrumentacion-campo/analisis-agua-liquidos-industria/turbidimetros-sensores-turbidez-agua>
- [25] <http://panamahitek.com/sensor-ultrasonico-hc-sr04-arduino/>

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA.

TRABAJO FIN DE GRADO

***< PECERA AUTOMATIZADA, MONITORIZADA
POR PANTALLA LCD Y POR MOVIL VIA
BLUETOOTH, BASADA EN LA PLATAFORMA
ARDUINO >***

DOCUMENTO 5- <ANEXOS>

Alumno/Alumna: <Salas, Jiménez, Héctor Javier>

Director/Directora (1): <Oleagordia, Aguirre, Iñigo Javier>

Curso: <2019-2020>

Fecha: < 28, Octubre, 2019>

4.1 ANEXOS

4.1.1 PLACAS PCB

A continuación, se van a adjuntar los diseños de algunos circuitos para ser impresos en placas PCB. Se presenta el diseño inicial y las capas SST-TOP-BOT. Los diseños se han realizado con el programa Orcad Layout. Y se ha usado parte de la biblioteca interna. En algunos otros casos, debido a que se trataban de un material muy concreto, se han tenido que crear los diseños para adjuntarlos luego en la creación de su respectiva layout.

ELECTRO-VÁLVULAS

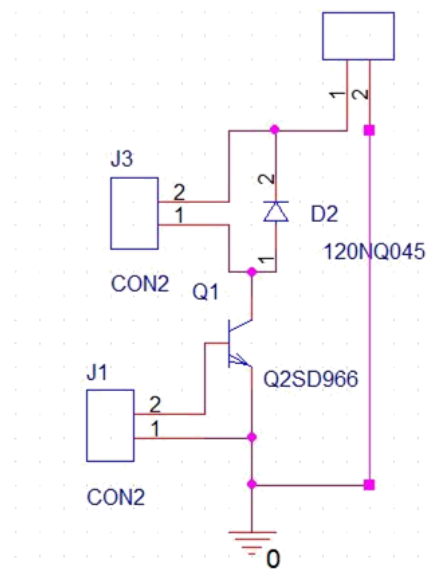


Fig66. Esquema circuito electro-válvulas.

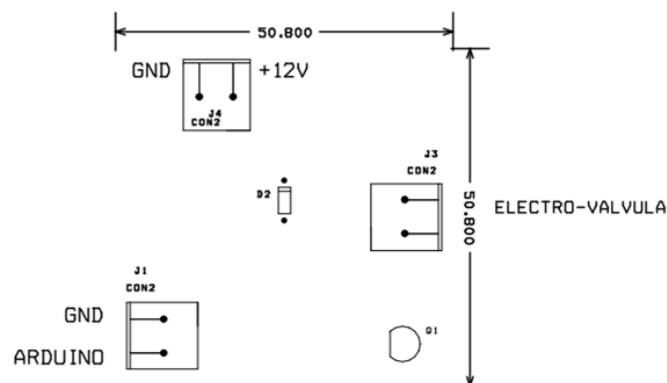


Fig67. Layer SST.

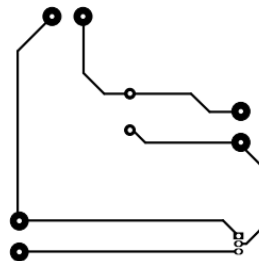


Fig68.. Layer BOTTOM.

SENSORES DE NIVEL

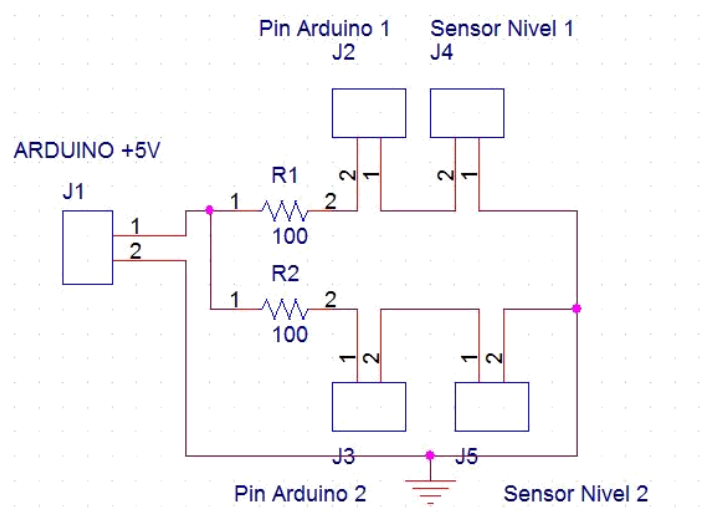


Fig69. Esquema circuito sensores de nivel.

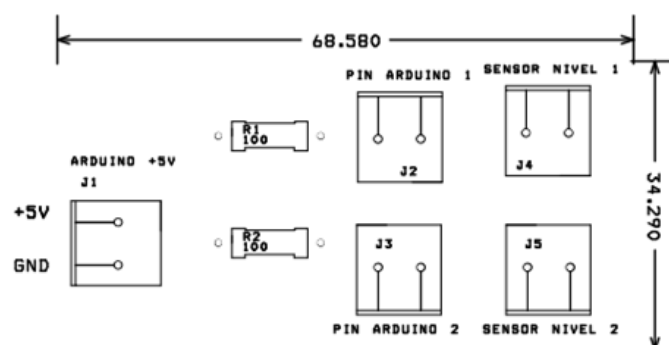


Fig70. Layer SST.

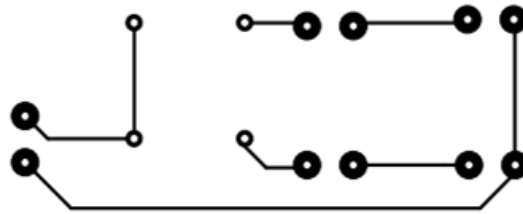


Fig71. Layer BOTTOM.

SENSORES NTC

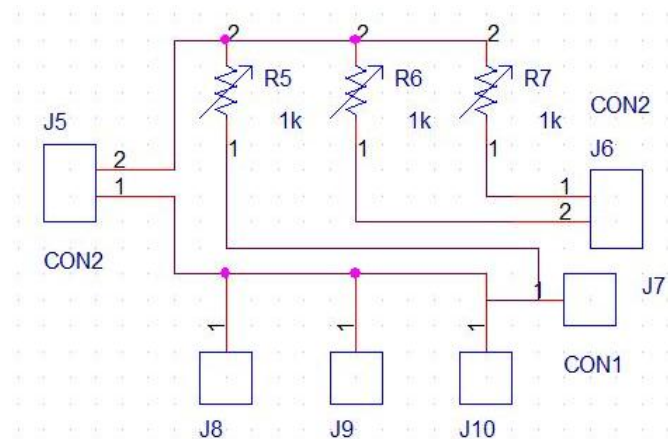


Fig72. Esquema circuito sensores NTC.

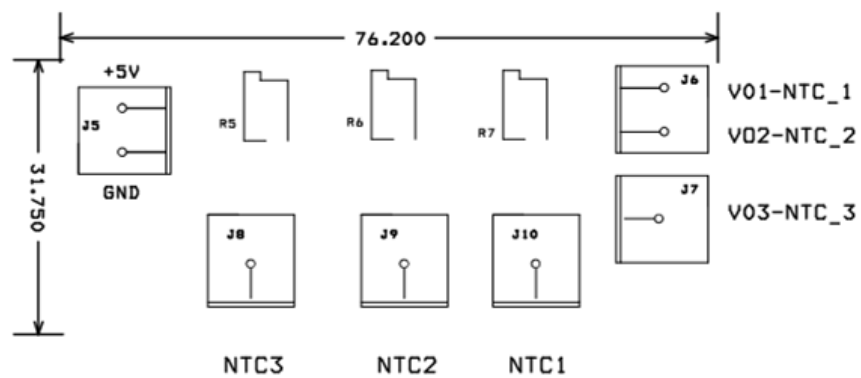


Fig73. Layer SST.

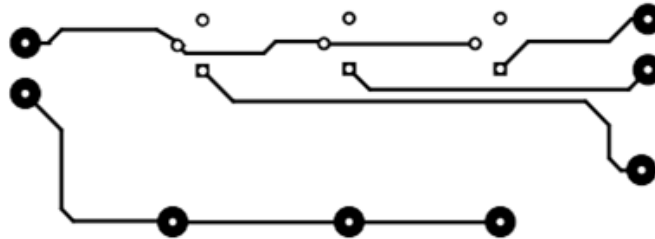


Fig74.. Layer BOTTOM.

SENSORES LM335

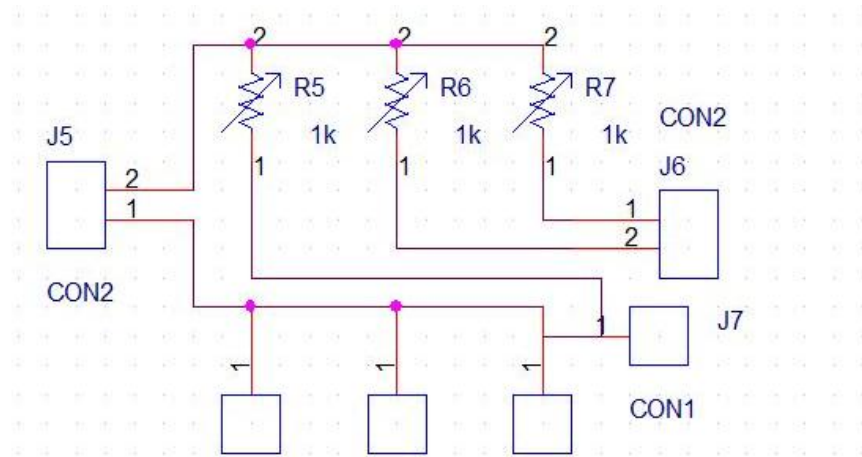


Fig75. Esquema circuito sensores LM335.

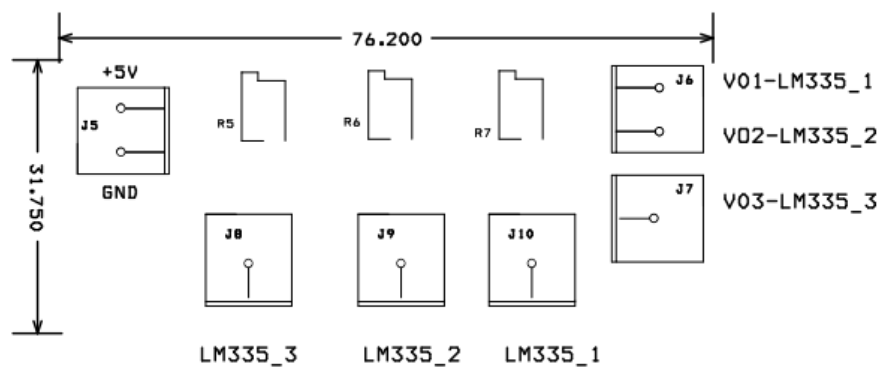


Fig76. Layer SST.

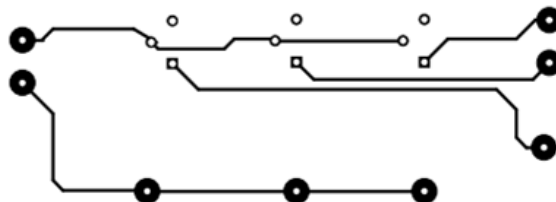


Fig77.. Layer BOTTOM.

RTD-PT100

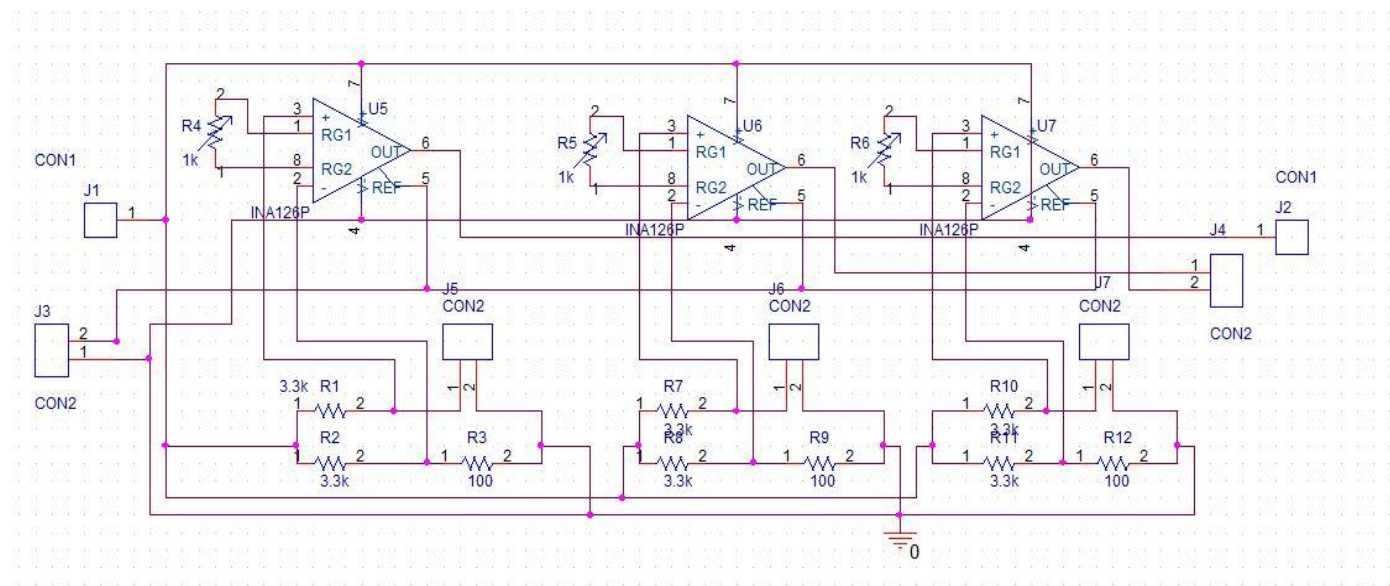


Fig78. Esquema circuito sensores PT100.

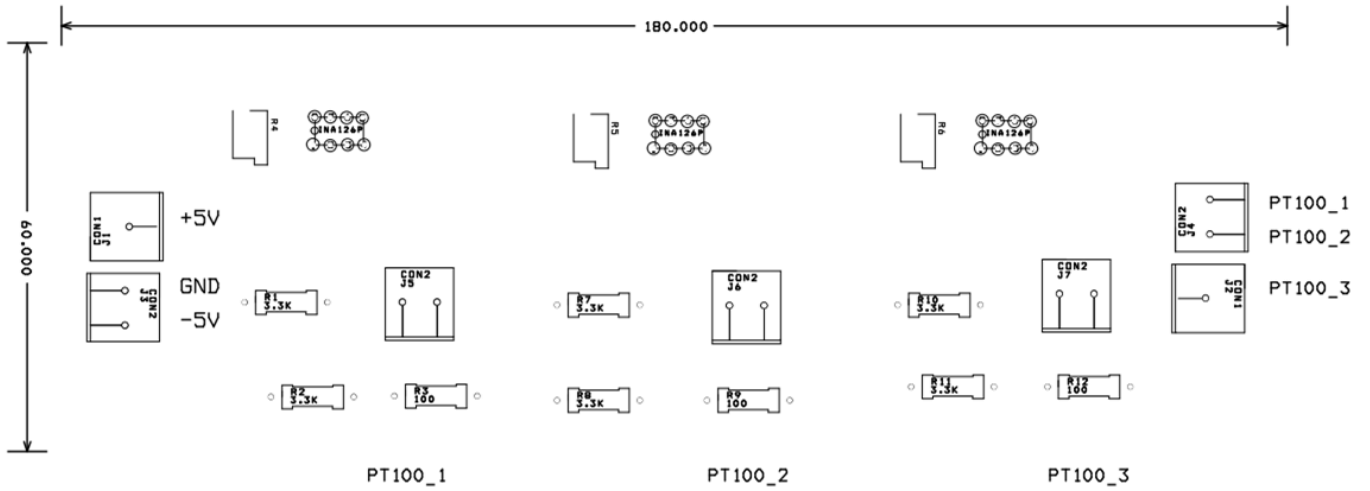


Fig79. Layer SST.

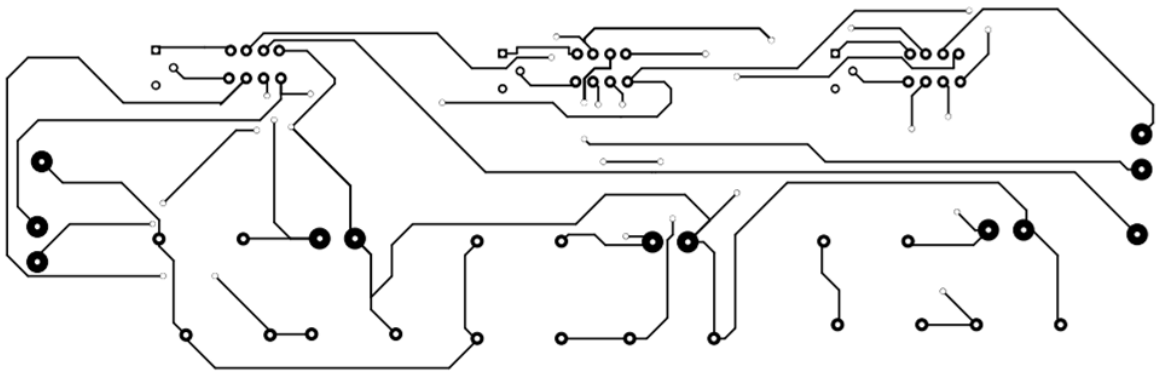


Fig80.. Layer BOTTOM.